EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

Instituto Politécnico Nacional
"La Técnica al Servicio de la Patria"

Research in Computing Science

Vol. 150 No. 11
November 2021

Research in Computing Science

Research in Computing Science, Año 20, Volumen 150, No. 11, noviembre de 2021, es una publicación mensual, editada por el Instituto Politécnico Nacional, a través del Centro de Investigación en Computación. Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, Ciudad de México, Tel. 57 29 60 00, ext. 56571. https://www.rcs.cic.ipn.mx. Editor responsable: Dr. Grigori Sidorov. Reserva de Derechos al Uso Exclusivo del Título No. 04-2019-082310242100-203. ISSN: en trámite, ambos otorgados por el Instituto Politécnico Nacional de Derecho de Autor. Responsable de la última actualización de este número: el Centro de Investigación en Computación, Dr. Grigori Sidorov, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othon de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738. Fecha de última modificación 01 de noviembre de 2021.

Research in Computing Science, year 20, Volume 150, No. 11, November 2021, is published monthly by the Center for Computing Research of IPN.

Volume 150(11)

Advances in Artificial Intelligence and Industry 4.0

Edgar Gonzalo Cossio Franco
Juan Humberto Sossa Azuela
María Elena González Bañales (eds.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2021

ISSN: in process

Formerly ISSNs: 1870-4069, 1665-9899

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

http://www.rcs.cic.ipn.mx
http://www.ipn.mx
http://www.cic.ipn.mx

Indexed in LATINDEX, DBLP and Periodica

Electronic edition

ISSN 1870-4069

Table of Contents

Page

ISSN 1870-4069

Earthquakes Insights and Predictions in Mexico Using Machine Learning

Adolfo Bustillos, Carmen Santiago, Gustavo Rubín

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
Mexico

juadolfob@gmail.com, {marycarmen.santiago,
gustavo.rubin}@correo.buap.mx

Abstract. Earthquake prediction is a challenging area of research and modest efforts have been made using machine learning for this application. In this work, seismic characteristics are calculated using seismological concepts, such as the Gutenberg-Richter law, rate of seismic change, anticipation frequency, release of seismic energy, and total recurrence time. With this, an Artificial Neural Network (ANN) classification model was built using sliding windows (as opposed to fixated sets of windows seen in another studies) to calculate its instances that were then resampled as a measure to counter the unbalanced data for the purpose of obtaining earthquake predictions; the model predicts more seismic events above a threshold without significantly sacrificing the metrics.

Keywords: Earthquake prediction, machine learning, neural network.

1 Introduction

Like many natural disasters, the earthquake causes many damages, economic and human losses and injuries [1]. That is why one of the most ambitious objectives of seismology is the prediction of earthquakes in the short term. In the mid-1970s, seismologists were confident that short-term earthquake prediction would be achieved in a short period of time. This confidence arose in part as a result of the first successful prediction of a major earthquake, the 1975 M7.4 Haicheng earthquake in China. Due to this prediction, an alert was issued within the 24-hour period before the main shock, probably preventing a greater number of casualties than the 1,328 deaths that actually occurred from this event. However, the impossibility of predicting another devastating earthquake 18 months later The 1976 Tangshan earthquake M7.8 was a major setback for the earthquake prediction effort. The victims of this earthquake number in the hundreds of thousands [2]. Lomnitz provides a summary of these events, as well as other successes and failures in earthquake prediction [3].

Earthquake prediction is not a task expected to be solved soon, but efforts of improving the methodology such as the ones in this study and those in the related

literature make the task closer to reality. In this study a new approach to calculate seismic indicators from a catalog is presented, with sliding windows (this approach is based on the fixed window approach that other studies use) hoping this will improve sensitivity and the quantity of instances, and it in addition gives solutions to the problem of unbalanced data. The always improving of seismic networks has permitted a huge amount of data, and thus has renewed hope that earthquake forecasting could be a feasible task, if combined with the improving of forecasting methods.

In Section 2 Related Literature and the seismic precursors using in them will be discussed, then in Section 3 the process of preparing the catalog and insights about mexican región will be described, in Section 4 there will be further details on seismic indicators and how they were computed in this research, at Section 5 multiple models of prediction, its performance and description of them will be presented, and at the end in the Conclusion Section the results will be discussed.

2 Related Literature

In the literature, there are different research methodologies, indicators and seismic precursors that have been used together with Machine Learning (ML) techniques for seismic analysis. These methodologies assume that there are variations in seismic precursors during pre-earthquake, and that there are patterns in these variations.

In 2007 and 2009, two important papers that complement each other were published by A. Panakkat and H. Adeli, first in [20] they formulated the prediction problem as a classification task, this attempted to predict the magnitude of the largest seismic event in a time window and a predefined region in the next month. They proposed eight of the so-called seismicity indicators: mathematically calculated characteristics, which can be used to assess the seismic potential of a region [16]. And later in [27] the same authors proposed the architecture of a probabilistic neural network (PNN) as a solution for the same problem that was formulated in [16] where they used the same set of seismicity indicators as input data for the network training. The model was tested with data from the Southern California seismic zone and yielded good prediction accuracies for events of magnitude 4.5 to 6.0. However, PNN did not work satisfactorily for earthquakes of magnitudes greater than 6.0.

In 2013 the article [21] was published, it used an ANN model to predict the magnitude of the earthquake in an interval or limited by a threshold during the next five days. The application of statistical tests and experiments showed the higher success rate of this method than other machine learning classifiers at the time [18]. The system is designed to provide two types of predictions: a) the probability of an earthquake occurring greater than a threshold magnitude in five days and b) the probability of a seismic event occurring within a predefined magnitude range [25].

In this study, new seismic parameters were defined based on Bath's law and Omori-Utsu's law, which describe the relationships between the main shock and aftershocks, according to their magnitude and frequency of occurrence, respectively.

In 2018, adding to the earthquake prediction problem, the earthquake prediction system (EPS) called EP-GP Boost was described in article [14]. This system is a classifier based on a combination of genetic programming (GP) and a boost algorithm called AdaBoost. A total of 50 features were calculated. The study of the applicability

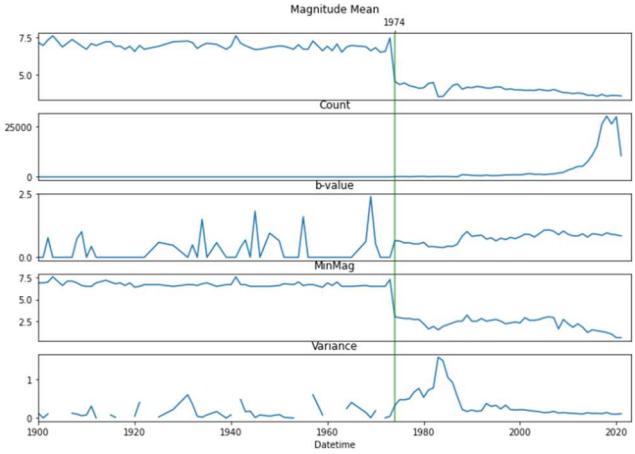


Fig. 1. Catalog perspective in Mexico, before 1974 the catalog is incomplete for magnitudes m^* > = 3.9, the gaps in variance, zeros and outliers in "b-value" are caused by lack of data. The variance graph is according to the magnitudes of the year, and the b-value graph describes the frequency of the earthquake size distribution based on the Gutenberg-Richter law.

of EP-GP Boost was carried out using data from previously used seismic zones, namely Chile, Hindukush and southern California. Experiments have shown outstanding performance in all three observed regions, both in terms of a low false alarm rate.

The best results were obtained for the Southern California region due to the completeness and quality of the corresponding earthquake catalog. However, the results for all regions show an improvement compared to previous studies [16, 18, 23].

Later the same year, [16] was written by the same authors as [14]. In this article, Asim et al. also used the approach for the use of seismicity indicators proposed in [14]. This time, 60 seismic parameters were calculated using various seismology concepts. As in their previous research, the authors intend to predict earthquakes of magnitude equal to or greater than 5.0 over the next 15 days. The system is a combination of different machine learning algorithms, and in each step, an algorithm uses the knowledge gained through learning from a previous one. They made use of mRMR, SVR, HNN and an EPSO for the same regions as [14]; it gave better results in the metrics.

In the region of Mexico there are project proposals of earthquake prediction using ML as [26] where random forest and deep learning models were proposed; also other damage prediction and modeling studies.

3 Earthquake Catalog Preparation

This data used in this study is based on earthquake catalogs. The catalogs were downloaded from the SSN (Servicio Nacional Sismológico) using the period from January 1900 to May 2021, the catalog contains 204806 seismic events.; They are available publicly, after discarding irrelevant columns and eliminating events with incomplete information as part of the cleaning and preprocessing we were left with

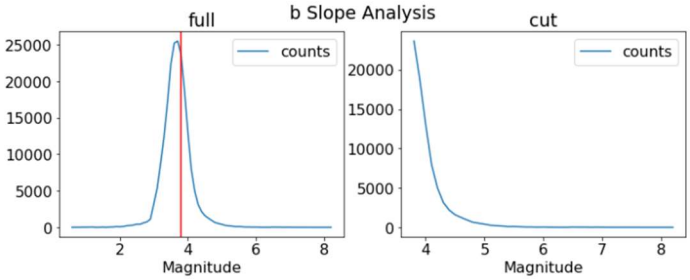


Fig. 2. Gutenberg-Richter Law applied in Mexico, from 1974 and for m^* > = 3.9 with a = 8.326 and b = 0.998, the y-axis contains the number of events, and the x-axis the magnitude of th events.

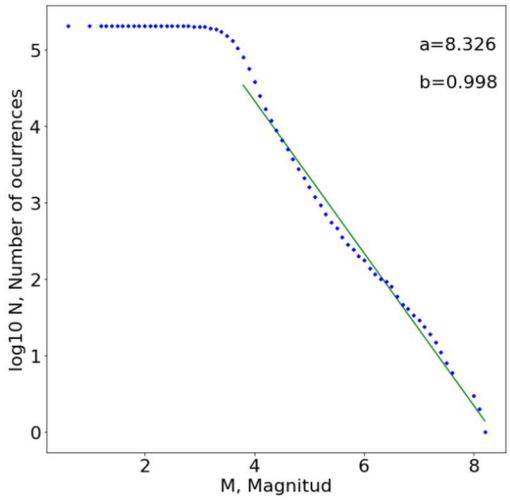


Fig. 3. Gutenberg-Richter Law applied in Mexico, starting in 1974 for m^*> = 3.9 with a = 8.326 and b = 0.998, it is calculated using the method of "linear least square regression" (lsq).

204806 seismic events and with their Date, Magnitude, Latitude and Longitude features.

3.1 Mexico Catalog Overview

The magnitude of completeness is defined as the lowest magnitude in which all earthquakes are successfully detected within a region and time period [5, 6], it is essential to analyze the seismicity of a region [6]. The magnitude of completeness varies in time and space and depends on many factors that affect the detection capacity of a seismological network, such as: the density and distribution of seismic stations, the type of instrumentation used, the efficiency of informing data from stations to the processing center, earthquake detection practices and procedures, among others [6, 7, 8, 9].

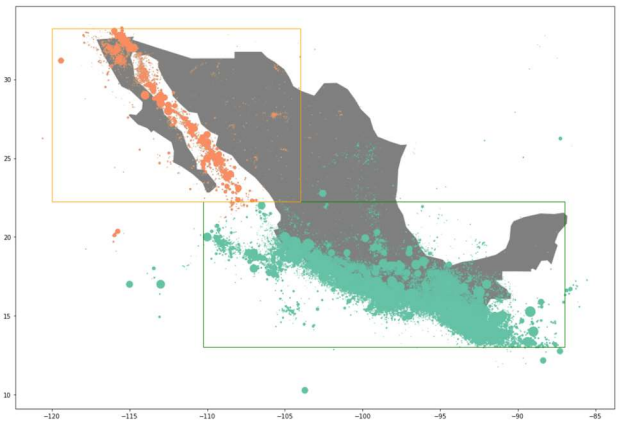


Fig. 4. Catalog division by KNN in function to its geometry, and selection of regions.

To satisfy the completeness of a region, the geometric extension is delimited, a temporal threshold and a cut-off magnitude is defined, the events that occurred below the thresholds and outside the boundaries of the region will not be used for the analysis, if the the catalog contains all the seismic events that occurred in a region, of magnitudes greater than or equal to as of a certain date, then the catalog is considered to be complete.

The seismic catalog of Mexico is probably complete for $m^*>$ = 7.7 as of 1846, $m^*>$ = 7.0 as of 1856 and $m^*>$ = 4.3 as of 1969 [10]. But the quality of the catalog increases dramatically from 1974, as shown in Fig. 1 and completeness is satisfied with $m^*>$ = 4.1, as shown in Fig. 2. This is the minimum date we will use to analyze the Mexico seismic catalog. The Gutenberg-Richter law describes the frequency of the earthquake size distribution [11], the law is described with the following expression:

$$log_{10}N = a - bM, \tag{1}$$

where a represents the total seismicity rate of the region, b is the relative distribution of the size of earthquakes and N is the number of earthquakes [12]. High b values mean a predominance of small earthquakes; conversely, a low b-value means that large earthquakes dominate smaller earthquakes.

Variations in the b-values both spatially and temporally are generally considered as clues for the precursors of large earthquakes [13]. To determine the cut-off magnitude in this study, "b-slope analysis" was used, the number of events for each magnitude are counted and the point where the curve deviates from the exponential behavior is selected, this is called the cut-off magnitude m^* and only events greater than or equal to this magnitude are considered for the analysis. When applying this analysis in the Mexican catalog for events from 1974 onwards, the resulting cut-off magnitude is m^* > = 3.9.

When applying the Gutenberg-Richter law in the Mexican catalog for m^* >= 3.9 from 1974, a value for a = 8.326 and b = 0.998 resulted, it is visualized in Fig. 3.

3.2 Geometric Extension of Catalog

The geometric extension of the catalog is from latitude 33.5 to 10.3 and longitude -120.5 to -85.5, that is, it only contains events within these limits, it covers the entire Mexican territory and the events are shown in Fig. 4, which is too extensive compared to other regions of analysis in related literature such as [14, 15, 16].

To select a smaller region, the catalog was separated into two parts using the KNN algorithm according to its geometry. Fig. 4, this measure also increases the correlation of the data; After a longitude and latitude range was arbitrarily selected to formalize the space of the analysis region, this facilitates replications of the analysis and comparison with other analyzes.

The resulting division between the northwest region and the southern region is consistent with the boundaries of the interacting tectonic plates in Mexico. The northwest region is represented in Fig. 4 in the upper part. covers the North Pacific, Gulf of California and Northwest Mexico, this encompasses longitudes from 120 to -104 and latitudes from 33.25 to 22.25, it contains 12,184 events where a = 7.313 and b = 0.988, and the adjacent plates are the North American Plate and the Pacific Plate. The southern region is represented in Fig. 4 in the lower part, it encompasses the tropical south Pacific and the west, center and southeast of Mexico, this encompasses longitudes from -110.25 to -87.5 and latitudes from 22.25 to 13, it contains 192622 events where a = 8.221 and b=0.987 and the adjoining plates are the North American Plate, Rivera Plate, Cocos Plate and Caribbean Plate.

3.3 Temporal Extension of the Catalog

The model is sensitive to the frequency with which events are reported, so it is necessary that the frequency of each year is similar. The largest number of reported events are in the most recent years (its general catalog quality also increases), and they become regular frequencies after 2010 for the northwest region and after 2017 for the southern regions as seen in Fig. 5.

For the Northwest region 2010 was used for the cutoff year, as quality improves for more recent dates, resulting in higher concentration and quality of events, and for the southern region, 2017 was used as the limit year, as of 2010 the frequency of reported events increases but until 2017 the frequency of events stops growing and stabilizes, in addition, the largest number of events with a greater quality is concentrated here.

3.4 Cut-Off Magnitude

The cutoff magnitude corresponds to the minimum magnitude in the catalog above which the earthquake catalog is considered complete, that is, there are no missing seismic events, and it has to be calculated for each region and time range. The value of the magnitude of cut depends on the integrity of the catalog, which itself depends on the instrumentation. Better instrumentation in a region leads to better catalog integrity with a low cutoff magnitude.

To evaluate the cut-off magnitude of this catalog we used an analysis of the Gutenberg-Richter curve [17]. The point where the curve deviates from the exponential

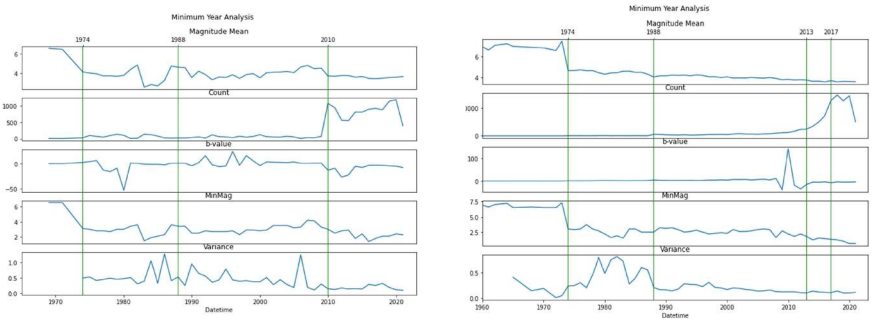


Fig. 5. Minimum year analysis for the northwest region on the left side and for the southern region on the right side.

behavior is selected as the cutoff quantity. All events below the cut-off magnitude are discarded and are not used in the analysis.

The magnitudes of earthquakes and frequencies of occurrences of a region are plotted as shown in figure Fig. 6. In the left sub-figure of each figure the red line marks the selected cut-off magnitude, in both northwest regions and south, with cut-off magnitude, m^*> = 3.9.

The resulting curves after cutting are shown in the right sub-figure of each figure; they follow an exponential behavior, which ensures that each catalog is complete at its respective cutting magnitude.

4 Calculation of Seismic Parameters

Indicators are the most important part of a classification problem [16]. Two types of indicators are calculated, parametric and non-parametric indicators that in several cases have multiple values based on different variations of a parameter, these relationships are called indicators or seismic parameters, the relationships that are used in this model are the geophysical and seismological parameters relevant for the prediction of earthquakes available in contemporary literature, these criteria are taken in order to retain the maximum information available on the internal geological state of the soil.

Time series prediction can be framed as a supervised learning problem. A sliding or moving window is a technique used to frame a set of time series data.

To calculate the metrics for this model, a multivariate of size $n \times 2$ moving window that looks at the date and magnitude of each catalog event slides through the catalog events and pulls subsets of the catalog of size $n \times 2$ at all positions in the window. The resulting $E - n + 1$ subsets will be subsets of size $n \times 2$, where E is the total number of events in the catalog.

The indicators are calculated for each subset generated, this results in $E - n + 1$ subsets of size $1 \times I$ where I is the number of indicators and labels (which are the indicators to be predicted) calculated.

The model will use the indicators to predict the labels, in this study the target parameters are called labels, which are the indicators that are sought to be predicted. To calculate the indicators and labels of new seismic events, it is not necessary to wait for

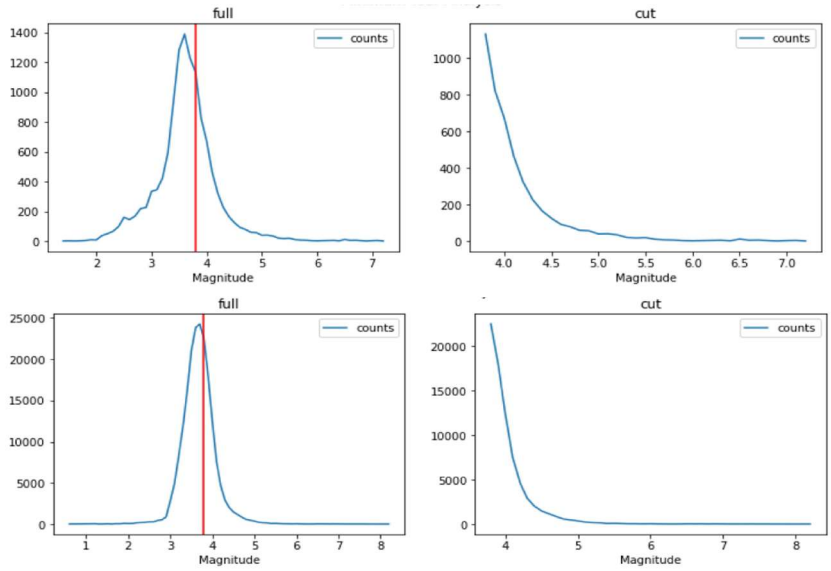


Fig. 6. Analysis of the Gutenberg-Richter curve for the northwestern region on the upper side and for the southern region on the lower side.

other new n events to occur, since the new J events and the latest n-j events can be used. In this study, the earthquake prediction problem is modeled as a binary classification problem, which seeks to predict whether a seismic event greater than or equal to the magnitude m will occur in the next d days. When using classification, the result is one class, among a limited number of classes.

With classes we refer to arbitrary categories according to the type of problem, in this study the classes are class 1 and class 0, if the model predicts that a seismic event greater than m will occur in the next d days the class will be 1, otherwise If the model predicts that a seismic event of magnitude greater than m will not occur in the next d days then the class will be 0. Although this classifier model can be trained to predict different magnitudes, there are models that use "Hybrid Machine Learning" techniques to predict the magnitude [18].

4.1 Parameters and Definitions

There are four parameters that must be determined to calculate the seismic indicators that the model is going to use.

First the parameter n is the window size, it describes the number of window events, the selected value is arbitrary and can be adjusted to obtain better results, normally in the related literature it is between 50 and 100, the size selected in this study is 100; The parameter m is the Magnitude threshold, this is the magnitude that will function as a threshold for classification, and several parameters and labels depend on this variable, the value normally used is between 5 and 6, the magnitude selected in this study is 5.5 (accommodating ourselves to the definition of a magnitude of "great earthquake" of [25]), we can't choose a very high value because the precision decreases in high

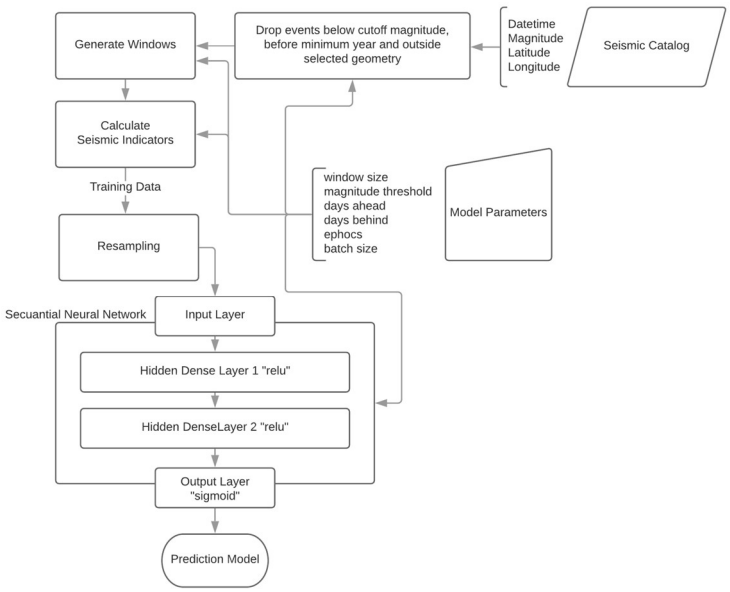


Fig. 7. It shows the structure of the prediction model. There are different variations of the model, where the Resampling stage is not used or is replaced by a Weighting stage, these variations are explained later.

magnitudes [25]; The parameter d_a is days ahead, this value can change depending on how many days ahead you want to predict the next seismic event, the value normally used is between 7 and 15, the value chosen in this study is 14. And the parameter d_b is days behind, the value normally used is 7, it is used to calculate the seismic indicators, the value chosen in this study is 7.

4.2 Indicators

The indicators used here are all those proposed in [20] and [21], plus a selection of the indicators proposed in [14] and [16].

The selected indicators are: date T_1 this is the date of the first seismic event of the window where T_i is the date of the i-th event of the window; Last date T_n last day of the window; Time elapsed T_Θ between the first and the last T_i of a window proposed in [20]; Events Time Difference Mean μ proposed in [20]; Standard deviation of the mean of the difference in elapsed time between events c [20]; Value a and b calculated numerically proposed in [20] using two different methods, one method represents linear least squares regression (lsq), and the other represents Maximum Maximum (mlk) as proposed in [14][16]; Mean magnitude M_{mean} proposed in [20]; Max Magnitude $M_{max\ observed}$ proposed in [20]; Expected Magnitude $M_{max\ expected}$ proposed in [20]; Magnitude Deficit ΔM proposed in [20]; Standard Deviation of the b-value σb proposed in [21]; Mean Square Deviation η proposed in [20]; Square Root Rate of Energy $dE\frac{1}{2}$ proposed in [20], if the release of energy stops, the phenomenon is known

Table 1. Result of the evaluation metrics.

Métricas	Baseline	Weighted	Resampled
TP	2043	2695	3000
FP	455	1617	946
TN	8221	7059	7730
FN	2170	1518	1213
Accuracy	0.79	0.75	0.83
Precision	0.81	0.62	0.76
Recall	0.48	0.63	0.71
F1 score	0.60	0.63	0.73
NPV	0.79	0.82	0.86
R score	0.43	0.45	0.60
MCC	0.51	0.45	0.61
FPR	0.94	0.81	0.89

as quiescence, which can be a precursor to a large seismic event [16]; Seismic Rate Change B β proposed in [10]; Seismic Rate Change Z z proposed in [19] but the mathematical expression used is in [28]; Maximum Magnitude in the last d days $x6$ prior to the last window event proposed in [21] ;and the Probability of event greater than m $x7$ proposed in [21].

Every indicator that made use of a and b values as parameters has two variations, one using a_{lsq} and b_{lsq} values and the other using a_{mlk} and b_{mlk} variation, there are 23 seismic indicators calculated in total.

4.3 Labels

It is still very ambitious to use this type of analysis to predict a magnitude (ie y1), in this study it is analyzed whether a magnitude is above or below a threshold magnitude (ie y2), y2 is 1 if the magnitude is above the threshold and 0 if the magnitude is below the threshold.

The precision of the analysis is lower with high magnitudes, since it is more difficult to predict events with higher magnitudes.

The proposed label is $y2$ but we calculate first $y1$ the Maximum magnitude in the next d days is calculated, this when $T \in [T_n, Tn + d_a)$ and $d_a = days\ to\ look\ ahead$; and then use $y1$ to calculate $y2$ the occurrence event greater than m is calculated using the equation 2:

$$y_2 = \begin{cases} 1 & \text{if } y1 >= m \\ 0 & \text{if } y1 < m \end{cases} \tag{2}$$

5 Results

To make the prediction, a neural network was used and the 23 calculated indicators were introduced. The structure of the model can be seen in Fig. 7.

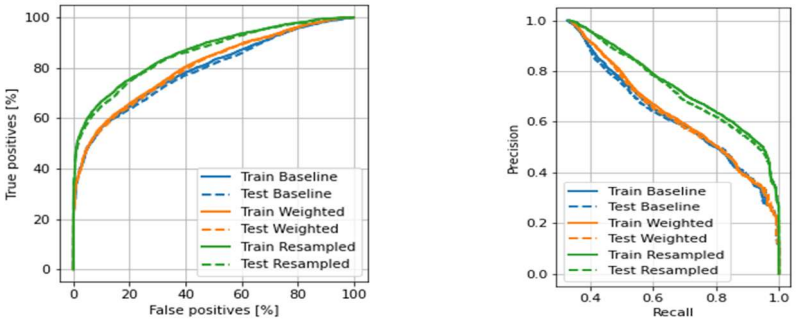


Fig. 8. The AUC curves on the left and the PRC curves on the right of the different models.

5.1 Neural Network

In this model, a sequential neural network was used. The neural network consists of two dense layers (deeply connected, meaning that each neuron in the dense layer receives information from all the neurons in its previous layer) with 23 neurons each hidden with "relu" activation, and one output layer with one neuron and sigmoid activation. There are 552 parameters for the first two layers and 24 for the third, adding up to a total of 1128 parameters.

To train a model on unbalanced data, it is fitted with a lot size large enough to ensure that each lot has a decent chance of containing some positive samples. Class 1 samples are called positive samples and class 0 samples are negative samples.

5.2 Metrics

Model performance was measured using the test sample obtained from 30% of the data, and training was performed on the training sample obtained from 70% of the data (20% of this sample was also used for validation), with the metrics, TP, FP, TN, FN, Accuracy, Precision, Recall, False Positive Ratio (FPR), Negative Predictive Value (NPV), MCC, F1 score and R score, Area Under the Curve (AUC), Precision Recall Area (PRC).

It is often possible to calibrate the parameters of a model and improve the results of one metric at the expense of another. The model has 3 variations that outperform each other in different metrics. Those models performed better on different metrics. The metrics of the models are shown in Table 1 and the models are explained later.

The purpose of analyzing the results through these mentioned metrics is that each performance metric highlights a certain aspect of the results. Therefore, the purpose is to highlight all the merits and demerits of the results obtained through the different variations of the proposed prediction models.

In the Baseline model the indicators are entered without modification, and there are no special parameters for the model. All models were trained using 200 epochs.

In the Weighted model the indicators are introduced without modifying, but in the model the weight parameter is added so that classes 1 and 0 have different weight when classifying. The objective is to identify earthquakes above the threshold, but according to the Gutenberg–Richter law we will not find many of those class 1 samples compared

to those of class 0, for this reason we must give more statistical value to class 1. With these Class weights, accuracy, and precision are lower because there are more FP, but recall and AUC are higher because the model also found more TP.

Despite having a lower precision, this model has a higher recall (and identifies more earthquakes above the threshold). To calculate the weights for the class 0 it was used the expression $weight\_0 = (1/neg) * (total/2.0)$ resulting in a weight value of 0.74, and for class 1 the following $weight\_1 = (1/pos) * (total/2.0)$ resulting in a weight value of 1.54 In the Resampled model the indicators are modified, and the parameters were also modified.

The approach is related to the previous one, but instead of calculating weights for each class, a resampling of the data set is performed on the minority class and thus balancing the data set, the result will be approximately 50% of data class 1 and 50 % of data class 0. The number of steps per epoch was modified, this is the number of batches needed to see each negative example once. With this, instead of class 1 data being displayed in one batch with a large weight, it is displayed in many different batches with a small weight. This smoother gradient signal makes training the model easier.

The PRC and AUC curves of every model and their variations can be seen in Fig. 8.

Metrics from different studies are not directly comparable, in contrast to other studies here we use a sliding window of a fixed size to calculate the instances that will be introduced to the model, other studies divide the dataset in chunks of a fixed size to calculate the instances, this results in almost as half as instances and less sensitivity to the indicator changes.

Facing the problem as a classification problem using time series and a sliding window, is the model proposed, and treating it also as a problem with unbalanced data gave better results after using the weighted model and in greater degree the resampled model, in comparison to the baseline model.

There are downsides from using the resampling model, first it takes more time and memory to train, second as it resamples the data, it runs the risk of having an overfitting problem.

6 Conclusion

In this interdisciplinary analysis, earthquake prediction has been performed through the interaction of earthquake precursors and computational Machine Learning techniques. The Resampled model with a performance from 0.60 to 0.89 in different metrics, is the most suitable model for this type of prediction since it predicts more seismic events above the threshold without significantly sacrificing the other metrics, giving the most desirable results of all three models.

The performance of the resampled model in the company of using sliding windows made this comparable with that of other techniques in the other studies. In the future more progress will have been made and more data will be available, machine learning has proved to be very useful in many fields, and many efforts are being made to adequate the use of machine learning for earthquake prediction.

References

1. Asim, K. M., Martínez-Álvarez, F., Basit, A., Iqbal, T.: Earthquake magnitude prediction in Hindukush region using machine learning techniques. Hazards, vol. 85, no. 1, pp 471–486 (2017) doi: 10.1007/s11069-016-2579-3
2. Cicerone, R. D., Britton, J., Ebel, J. E.: A systematic compilation of earthquake precursors. Tectonophysics, vol. 476, no. 3–4, pp 371–396 (2009) doi: 10.1016 /j.tecto.2009.06.008
3. Lomnitz, C.: Fundamentals of earthquake prediction. John Wiley & Sons, New York, pp 326 (1994)
4. Géli, L., Piau, J. M., Dziak, R., Maury, M., Fitzenz, D., Coutellier, Q., Henry, P.: Seismic precursors linked to super-critical fluids at oceanic transform faults. Nature Geoscience (2014) doi: 10.1038/ngeo2244
5. Mignan, A., Woessner, J.: Estimating the magnitude of completeness for earthquake catalogs. Community Online Resource for Statistical Seismicity Analysis, no. 4, pp 1–45 (2012)
6. Arroyo, M., Godínez, K., Linkimer, L.: Completitud del catálogo de la red sismológica nacional de Costa Rica. Boletín de Geología, Universidad Industrial de Santander Bucaramanga, Colombia, vol. 39, no. 3, pp. 87–98 (2017)
7. Chouliaras, G.: Investigating the earthquake catalog of the National Observatory of Athens. Natural Hazards and Earth System Sciences, vol. 9, pp 905–912 (2009)
8. Mignan, A., Woessner, J.: Estimating the magnitude of completeness for earthquake catalogs. Community Online Resource for Statistical Seismicity Analysis, vol. 4, pp 1–45 (2012)
9. Wiemer, S., Wyss, M.: Minimum magnitude of completeness in earthquake catalogs: Examples from Alaska, the Western United States, and Japan. Bulletin of the Seismological Society of America, vol. 90, no. 4, pp 859–869 (2000)
10. Rudolf-Navarro, A., Muñoz-Diosdado, A., Angulo-Brown, F.: Seismic quiescence patterns as possible precursors of great earthquakes in Mexico (2010)
11. Gutenberg, B., Richter, C.: Frequency of earthquake in California. Bulletin of the Seismological Society of America, vol. 34, no. 4, pp 185–188 (1944)
12. Wu, Y. M., Chen, S. K., Huang, T. C., Huang, H. H., Chao, W. A., Koulakov, I.: Relationship between earthquake b-values and crustal stresses in a young orogenic belt. Geophysical Research Letters, vol. 45, pp 1832–1837 (2018) doi: 10.1002 /2017GL076694
13. Smith, W. D.: The b-value as an earthquake precursor. Nature, vol. 289, no. 5794, pp 136–139 (1981). doi: 10.1038/289136a0
14. Asim, K. M., Adnan-Idris, I. T., Martínez-Álvarez, F.: Seismic indicators based earthquake predictor system using genetic programming and AdaBoost classification. Soil Dynamics and Earthquake Engineering, vol. 111, pp 1–7 (2018) doi: 10.1016/j.soildyn.2018.04.020
15. Asim, K. M., Martínez-Álvarez, F., Basit, A., Iqbal, T.: Earthquake magnitude prediction in Hindukush region using machine learning techniques. Natural Hazards, vol. 85, pp 471–486 (2016)
16. Asim, K. M., Idris, A., Iqbal, T., Martínez-Álvarez, F.: Earthquake prediction model using support vector regressor and hybrid neural networks. PLoS ONE, vol. 13, no. 7 (2018) doi 10.1371/journal.pone.0199004
17. Salam, M., Ibrahim, L., Abdelminaam, D.: Earthquake prediction using hybrid machine learning techniques. International Journal of Advanced Computer Science and Applications, vol. 12, no. 15. (2021) doi: 10.14569/IJACSA.2021.0120578
18. Marsan, D., Wyss, M.: Seismicity rate changes. Community Online Resource for Statistical Seismicity Analysis (2011) doi: 10.5078/corssa-25837590
19. Panakkat, A., Adeli, H.: Neural network models for earthquake magnitude prediction using multiple seismicity indicators. International Journal of Neural Systems, vol. 17, no. 01, pp 13–33 (2007) doi: 10.1142/S0129065707000890

20. Reyes, J., Morales-Esteban, A., Martínez-Álvarez, F.: Neural networks to predict earthquakes in Chile. Applied Soft Computing, vol. 13, no. 2, pp 1314–1328 (2013) doi 10.1016/j.asoc.2012.10.014
21. Matthews, M., Reasenberg, P.: Statistical methods for investigating quiescence and other temporal seismicity patterns. Pure and Applied Geophysics, vol. 126, pp 357–372 (1988)
22. Habermann, R.: Precursory seismic quiescence: Past, present, and future. Pure and Applied Geophysics, vol. 126, pp 279–318 (1998)
23. Morales-Esteban, A., Martínez-Álvarez, F., Reyes, J.: Earthquake prediction in seismogenic areas of the Iberian Peninsula based on computational intelligence. Tectonophysics, vol. 593, pp 121–134 (2013) doi: 10.1016/j.tecto.2013.02.036
24. Galkina, A., Grafeeva, N.: Machine learning methods for earthquake prediction: A survey. In: The Fourth Conference on Software Engineering and Information Management, Saint Petersburg, Russia (2019)
25. Telesca, L., Wen, Y.: Development of machine learning-based innovative methods of seismic forecasting in Italy and Mexico. Development of machine learning-based innovative methods of seismic forecasting in Italy and Mexico, Consiglio Nazionale delle Ricerche (2021)
26. Adeli, H., Panakkat, A.: A probabilistic neural network for earthquake magnitude prediction. Neural Networks, vol. 22, no. 7, pp 1018–1024 (2009) doi: 10.1016/ j.neunet.2009.05.003
27. Habermann, R. E., Wyss, M.: Reply [to "Comment on Habermann's Method for detecting seismicity rate changes"]. J Geophys Res, vol. 92, no. B9, pp 9446–9450 (1987) doi: 10.1029/JB092iB09p09446

ISSN 1870-4069

Evaluation of Models for Estimating Cardiovascular Risk based on the Framingham Table

Norma Karen Valencia[1], Edgar Corona Organiche[2], Abraham J. Jiménez Alfaro[2], Griselda Cortés Barrera[2]

[1] Tecnológico de Estudios Superiores de Chimalhuacán, Mexico

[2] Tecnológico de Estudios Superiores de Ecatepec, Mexico

karenvalencia@teschi.edu.mx, {ecorona,ajimenez,gcortes}@tese.edu.mx

Abstract. Cardiovascular diseases (CVD) are responsible for the majority of deaths in the world, approximately 30%; which makes it important to estimate the risk of developing cardiovascular disease in order to prevent and reduce this index. The main purpose of this work is to evaluate the performance of different algorithms to select which of them is the most appropriate to estimate cardiovascular risk. The present work presents the comparison of the estimators developed through decision trees, Naive Bayes algorithm, a vector support machine (SVM) and a Neural Network composed of six inputs, 8 hidden layers and an output to estimate Cardiovascular Risk; considering seven risk factors as input and three types of risk as output: high, intermediate, and low. The confusion matrix of each implemented algorithm shows the precision of each of the models, taking as a reference the cardiovascular risk stratification by means of the Framingham function, allowing to establish the degree of error of each one of them. The analysis of the results allows us to observe that the performance of the multilayer perceptron was below the Decision Tree and the Naive Bayes Algorithm, which provide very similar results. Therefore, decision trees are a good choice to determine the level of cardiovascular risk. Efficiently estimating cardiovascular risk will allow health centers to establish strategies to reduce this factor in their attended population.

Keywords: Decision trees, Framingham table, vector support machine, naive Bayes, neural network.

1 Introduction

The World Health Organization (WHO) defines Cardiovascular Disease (CVD) as all those conditions that generate disorders of the heart and blood vessels, including: coronary heart disease, which is described as a disease of the blood vessels that supply the heart muscle. Also included are cerebrovascular diseases: diseases of the blood vessels supplying the brain; peripheral arteriopathies: diseases of the blood vessels;

Table 1. Risk of the Cardiovascular Event in a period of 10 years.

Risk level	10-year risk estimate percentage.	Years of possible cardiovascular event
Under	<15%	Older than 7 years
Intermediate	15-20%	Over 4 years
High	> 20%	Less than 5 years

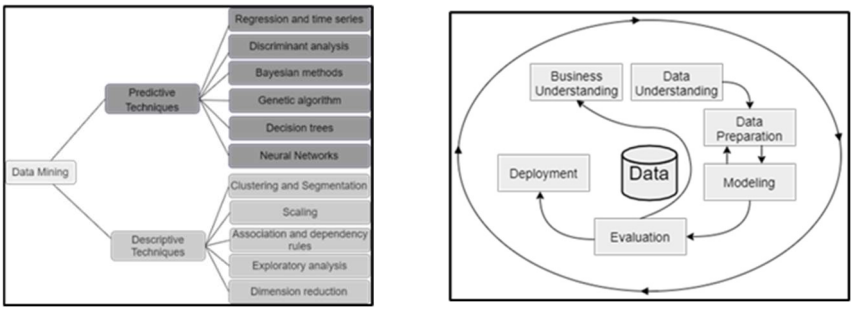


Fig. 1. a) Classification of Data Mining techniques, b) Phases of the CRISP-DM Methodology (GBD 2015 Risk Factors Collaborators., 2018).

rheumatic heart disease, congenital heart disease caused by malformations of the heart present from birth, adding to deep vein thrombosis and pulmonary emboli, which can detach (emboli) and lodge in the vessels of the heart and lungs [1].

In Mexico, it is estimated that CVD represents 20% of all deaths in adults. According to the National Institute of Statistics and Geography (INEGI), in 2016 136,342 deaths were reported due to heart disease, an increase of 7,611 deaths compared to 2015. Causes of death include ischemic heart disease, cerebrovascular diseases, hypertensive, among others [2].

The assessment of cardiovascular risks has generated various studies, since 2015 the Federal Government provided the "Risk Factors Questionnaire" to be used as an instrument to determine cardiovascular risk based on the risk stratification of the Framingham table. This work aims to verify if the decision trees generate better results for the Cardiovascular Risk Estimation.

2 Problematic

On September 25, 2015, the document entitled "Transforming Our World: the 2030 Agenda for Sustainable Development" was developed, adopted by member countries of the United Nations, including Mexico, and one of the objectives of this document is to develop strategies to prevent Non-Communicable Diseases, like CVD [3].

The study entity has historical records of more than 10 years, information concentrated in the Micro diagnosis certificates of its population. However, it does not have an information analysis tool that allows it to estimate or quantify morbidity, as well as mortality caused by cardiovascular diseases.

In addition to the above, the Health Institution does not have a model that allows it to estimate the risk of developing cardiovascular diseases in the population it serves.

Table 2. Description of Phases: CRISP-DM Methodology.

Phase	Description
Business understanding	It brings together the tasks of understanding the objectives and requirements of the project from a business or institutional perspective, in order to turn them into technical objectives and a project plan[11].
Understanding the Data	It includes the initial collection of data, with the aim of establishing a first contact with the problem[12].
Data Preparation	It consists of some tasks such as data selection by choosing a subset of the data collected in the previous stage. Cleaning the data, preparing them for the modeling phase, either by applying normalization techniques, discretization of numerical fields, treatment of null values, among others [13].
Modeling	It is suggested to make a comparison of the modeling techniques that are most appropriate to solve the problem[14].
Evaluation	The model is evaluated based on compliance with the success criteria of the problem, review the process followed taking into account the results obtained[15].
Implementation	This task takes the results of the evaluation and concludes a strategy for its implementation[15].

Likewise, it lacks the level of correlation that exists between the different risk factors that influence the development of this type of disease.

When prevention campaigns are started, the population is not fully captured, since the exhaustive analysis of data and risk factors for everyone is omitted, which generates an imprecise report of the population vulnerable to said diseases.

3 Theoretical Framework

The classic Framingham function estimates the risk of suffering an event in the next 10 years, considering death of coronary origin, non-fatal acute myocardial infarction, stable angina, or unstable angina (coronary insufficiency) as an event. The National Cholesterol Education Program in its latest document Adult Treatment Panel III (ATP-III) has modified this function to exclusively calculate the risk of so-called "hard" events, that is, non-fatal acute myocardial infarction and coronary death, excluding diabetics from this estimate by considering them directly at high risk. [4, 5].

Derived from the above, Table 1 summarizes the risks studied for cardiovascular events, the "Very High" risk level is not considered since it is included within the high risk, all these risks belong to the stratification based on Framingham table that is used in the Health Sector through the Clinical Practice Guide: Detection and Stratification of Cardiovascular risk factors [6].

The "Risk Factors Questionnaire" consists of the following variables to detect risk factors: Diabetes, Glycemia, Arterial Hypertension, Cancer (Colorectal, pulmonary, Oral, Gastric), cardiovascular diseases (heart disease, embolism, and hypertension), for the present study, those shown in Table1 are taken. The applications commonly developed with predictive analytics are predicting risks, predicting activation of new clients, predicting sales, among others. This type of analysis is characterized by

Table 3. Risk factors used in the models.

Risk factor	Neural Network	Naive bayes	Decision trees
Age	X	X	X
Diastolic pressure	X	X	X
Systolic pressure	X	X	X
Diabetes	X	X	X
Hypertension	X	X	X
Inheritance	X	X	X

Table 4. Cardiovascular risk probabilities using Bayes' theorem.

Risk factor	Risk of cardiovascular disease (RECV)
Age	$P(RECV\|E)$
Diastolic pressure	$P(RECV\|E \cap Pd)$
Systolic pressure	$P(RECV\|E \cap Pd \cap Ps)$
Diabetes	$P(RECV\|E \cap Pd \cap Ps \cap D)$
Hypertension	$P(RECV\|E \cap Pd \cap Ps \cap D \cap Hip)$
Inheritance	$P(RECV\|E \cap Pd \cap Ps \cap D \cap Hip \cap Her)$

Table 5. Review of variables in various cardiovascular risk stratification tables.

Variables	Identification card Microdiagnosis	SCORE	Framingham	PCE
Age	*	*	*	*
Sex	*	*	*	*
Cholesterol level	Only valued people	*	*	*
Pressure Systolic mmg	Only valued people	*	*	*
Smoking	*	*	*	*
Diabetes	*	*	*	*
Diet	*	*	*	*
Physical activity	*	*	*	*
Weight	*	*	*	*

requiring a training set, which is made up of a data log. Discrete prediction and continuous prediction tasks can be performed in predictive analytics [7].

Within data mining, classifiers can obtain the incidence of cardiovascular events. Bayesian classifiers allow to classify discrete and limited events (independent variables) in a certain number of classes by defining a statistical function for each class. In Fig. 1a, the graph is showing its classification. Bayesian networks are a probabilistic model by means of which it is feasible to construct a graph between the causes of an event (independent variables) and its consequences (dependent variables) [8].

4 Methodological Framework

There are several methodologies that provide a series of steps to follow in order to carry out a proper implementation of data mining. According to polls published in

KDnuggets [9], the most used methodologies are CRISP-DM (Cross Industry Standard Process for Data Mining), SEMMA, KDD and Catalyst. This information can be consulted in the link https://www.kdnuggets.com/2014/10/c risp-dm-top-methodology-analytics-data-mining-data-science-projects.html.

The CRISP-DM is a free distribution methodology that can work with any tool to develop any project, it structures the life cycle of a Data Mining project, its phases described in Table 2, interact with each other iteratively during the development of the project as shown in Fig. 1b designed in a neutral way to the tool used for the development of the project [10].

5 Methodological Development

5.1 Business Understanding or Understanding

The Framingham table works with nine risk factors as well as other risk stratification tables such as the Systematic Coronary Risk Evaluation (SCORE) that measures the risk of mortality due to cardiovascular disease at 10 years in a European population aged 40 to 65 years [16], the present study focuses on the first table and on the practice of clinical guidelines for cardiovascular risk stratification.

5.2 Understanding the Data

Nine risk factors that are presented in Table 3 were considered, three characteristics were discarded in the first place, a non-modifiable factor such as sex and two modifiable factors which are weight, smoking and physical activity. The risk factors that were not considered have a close relationship with those that were used as characteristics, in the case of the risk factor "weight" it is linked to systolic and diastolic pressure, as well as hypertension and diabetes.

5.3 Data Preparation

For the model to work optimally, the data must be standardized complying with the characteristics, three categories High, Intermediate and Low are established as output, from inputs six characteristics are taken containing a total of 248 records, the input data as well as the output are shown in Table 5 and Fig 2.

5.4 Modeling

The Neural network architecture consists of 6 inputs, 1 hidden layer with 8 neurons and three outputs as shown in Fig. 2a.

5.4.1 Neural Network

The essential characteristics of a neural network are the nodes (organized in layers), the network architecture, which describes the connection between the nodes, and the

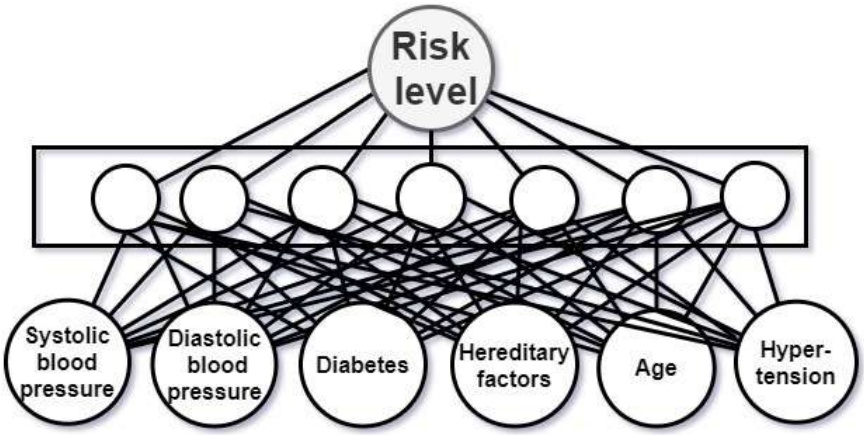


Fig. 2. a) Neural Network Architecture.

Table 6. Partial sample of the data entered the models (Hered. Mean Hereditary and Hyper. Hypertension).

TARGET	AGE	GLUCOSE	PD	PS	HERED.	HYPER.
HIGH	47	101	134	81	0	1
LOW	56	98	126	86	0	0
INTERMEDIATE	66	113	115	80	0	0
LOW	35	97	122	83	0	0

algorithm used to find the values of the network parameters (weights). [17] The layers of a network can be: a) To begin with, made up of neurons that introduce the six risk factors, no processing is performed in these neurons, b) Hidden made up of neurons whose inputs come from the previous layer and c) Output, formed by neuron that indicates the level of risk.

The optimization method used is Adam, which only requires first-order gradients with low memory requirements. The method calculates the individual adaptive learning rates for different parameters from the estimates of the first and second moment of the gradients; the name Adam is derived from the adaptive estimation of the moment [18]. The Orange Datamining software (version 3.26.0) was used as a data analysis tool. The Fig 3, shows the widgets used to work with the models.

5.4.2 Naive Bayes Algorithm

The proposed model uses the Naïve Bayes algorithm based on age, then other factors, generating the probability of suffering from cardiovascular diseases. The following table shows the probabilities obtained. Where: $P\ (RECV \mid E)$ = Probability of cardiovascular risk-taking Age as a risk factor. $P\ (RECV \mid E \cap Pd)$ = Probability of cardiovascular risk-taking Age and diastolic pressure as risk factors. $P\ (RECV \mid E \cap Pd \cap Ps)$ = Probability of cardiovascular risk-taking Age, distolic pressure and systolic pressure as risk factors. $P\ (RECV \mid E \cap Pd \cap Ps \cap D)$ = Probability of cardiovascular risk-taking Age, diastolic pressure, systolic pressure and Diabetes as a risk factor. $P\ (RECV \mid E \cap Pd \cap Ps \cap D \cap Hip)$ = Probability of cardiovascular risk-taking Age, diastolic pressure, systolic pressure, Diabetes and Hypertension as a risk factor. $P\ (RECV \mid$

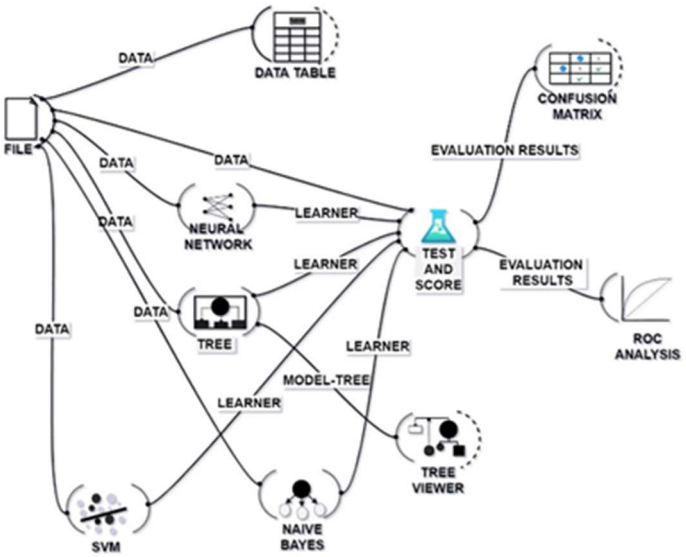


Fig 3. The RN configuration handles 8 hidden layers with logistic activation function and Adam algorithm with a maximum of 200 iterations.

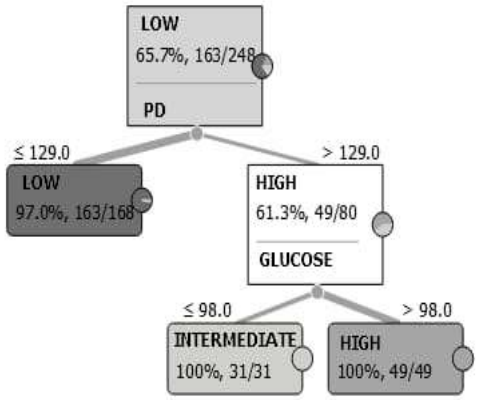


Fig. 4. a) Tree Architecture.

E∩Pd∩Ps∩D∩Hip∩Her) = Probability of cardiovascular risk-taking Age, diastolic pressure, systolic pressure, Diabetes, Hypertension and Heredity as a risk factor.

5.4.3 Classification Trees

Classification trees, which are an alternative to more classical statistical techniques, such as multiple regression, ANOVA analysis, logistic regression, discriminant

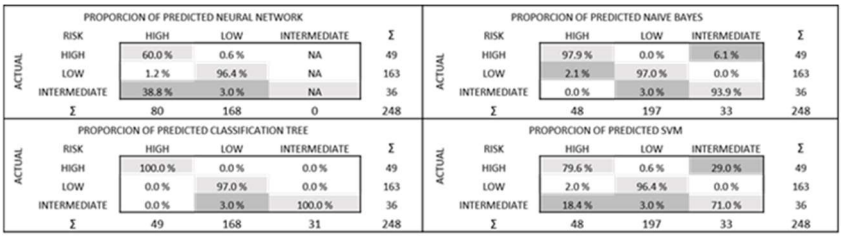


Fig. 6. Confusion matrix (misclassified is shown in a darker tone).

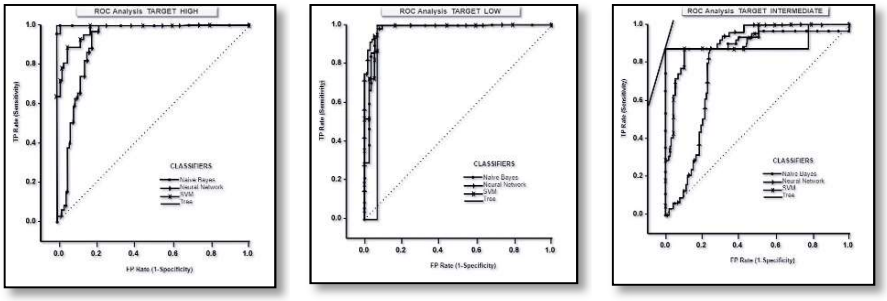


Fig. 7. Analysis of ROC Curves of the models: a) High Risk, b) Low Risk. c) ROC Curve Analysis of Intermediate risk models.

Table7. Model results.

Model	* AUC	Classification Accuracy	Precision	Specificity
Naive bayes	0.977	0.976	0.976	0.961
Neural Networt	0.944	0.843	0.745	0.907
SVM	0.943	0.875	0.871	0.930
Tree	0.955	0.980	0.980	0.961

* AUC = Area under ROC curve

analysis, and survival models. (Classification trees also seem to obtain better predictive rates than the rest of data mining techniques when mostly categorical information is used [19].

The configuration for SVM is Cost: 1.0, Regression Loss Epsilon 0.10, Kernel: Sigmoid ang Iteration Limit: 100. The parameters for the tree are Limit the maximal tree depth to: 100, Stop when majority reaches: 95 and Min. number of instances in leaves: 2.

5.4.4 Support Vector Machines

This model uses the technique of support vector machines (SVM), to make hyperplanes that divide the population between usual and unusual (in our study low, intermediate and high risk), from the training population, with which it is achieved classify the validation population [20].

6 Results

The data that were treated for this study were from 248 patients corresponding to a section of the studied Health Center. The configuration chosen with respect to the SVM is shown in Table 4, as it is observed the cost was chosen of 1 and the loss of 0.10. Access to training data is here[1].

The perceptron did not correctly identify the level of Intermediate Risk as seen in the confusion matrix Fig. 6, in such a way that it only worked on two levels of Risk: low and high, that is, its classification precision is 0.843 as seen in the data in Fig. 6.

The general results are shown in the table 6, in which the performance of the models is compared. A more global way of knowing the quality of the test in the full spectrum of cut-off points is using ROC curves (receiver operating characteristics), which constitute a fundamental and unifying tool in the evaluation process, see Fig. 7 [21]. The analysis had as parameters: Default threshold (0.5) point. Show performance line: FP Cost: 500, FN Cost: 500 and Prior probability: 66%.

From the above data, Table 6 is generated where the analysis of the ROC curves is interpreted numerically as well as the classification matrix corresponding to each model. The result of Analysis by means of the ROC curves shows that the trees generate better results.

7 Conclusions

Based on the results of the confusion matrices, it is concluded that the decision trees are the most suitable for estimating cardiovascular risk, followed by the Naive Bayes algorithm and the one with the lowest performance was the SVM. The efficient estimation of cardiovascular risk will allow health centers to establish strategies to reduce both this index and the mortality index in their attended population due to the development of cardiovascular disease.

References

1. OMS: Enfermedades cardiovasculares hechos clave (2021)
2. OMENT: Un panorama de las enfermedades cardiovasculares. Observatorio Mexicano de Enfermedades no Transmisibles (2018)
3. ONU: Informe de los objetivos de desarrollo sostenible 2020. Naciones Unidas (2020)
4. Pacheco, A. M., Jáquez, T. J.: Prevalencia de síndrome metabólico en la consulta externa. Rev Sanid Milit Mex, vol. 71, pp. 264–275 (2017)
5. Godala, M., Materek-Kuśmierkiewicz, I., Moczulski, D., Szatko, F., Gaszyńska, E., Kowalski, J.: Estimation of cardiovascular risk in patients with metabolic syndrome. Pol Merkur Lekarski; vol. 41, no. 246, pp. 275–278 (2016)
6. Secretaría de Salud. Intervenciones de enfermería para la prevención de complicaciones de Enfermedades Cardiovasculares en adultos en los tres niveles de atención. México, G.P.C. SS-365-16 (2016)
7. Lanzarini, L. C., Hasperué, W., Villa-Monte, A., Basgall, M. J., Molina, R., Rojas-Flores, L., Corvi, J. P., Jimbo-Santana, P., Fernández-Bariviera, A., Puente, C., Olivas-Varela, J.

[1] https://mega.nz/folder/F5U1DKKI#Ucryqpb4KAaVa4 IDQlt3lA.

A.: Minería de datos y Big data: Aplicaciones en riesgo crediticio, salud y análisis de mercado. In: XX Workshop de Investigadores en Ciencias de la Computación, pp. 350–354 (2018)
8. Castrillón, O. D., Sarache, W., Castaño, E.: Sistema bayesiano para la predicción de la diabetes. Inf. Tecnol., vol. 28, no. 6, pp. 161–168 (2017)
9. Condori, S. E.: Modelo de minería de datos para la predicción de casos de anemia en gestantes de la provincia de Ilo. Tesis Universidad Nacional de Moquegua, Perú (2019)
10. Mejía, H. R.: Modelo de minería de datos para la identificación de patrones que influyen en la mora de la cooperativa de ahorro y crédito San José S.J., M. S. Tesis, Pontificia Universidad Católica de Ecuador. Ecuador (2018)
11. Betancourt, S., Gómez, M. C., Quintero, J. B.: Inteligencia de negocios aplicada al ecoturismo en Colombia: Un caso de estudio aplicando la metodología CRISP-DM. In: XIV Congreso Ibérico de Sistemas y Tecnologías de la Información, CISTI 2019, Coimbra, Portugal (2019)
12. Huber, S., Wiemer, H. Schneidera, D., Ihlenfeldt, S.: DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model. In: 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, pp. 403–408 (2018)
13. Schäfer, F., Zeiselmair, C.: Sintetizar CRISP-DM y gestión de calidad: un enfoque de minería de datos para procesos de producción. In: Conferencia Internacional IEEE 2018 sobre Gestión de Tecnología, Operaciones y Decisiones, Marrakech, Marruecos (2018)
14. Espinosa, J. J.: Aplicación de metodología CRISP-DM para segmentación geográfica de una base de datos pública. Ing. en inv. Tec., vol. XXI pp. 1–17 (2020)
15. Mancilla, G., Leal-Gatica, P., Sánchez-Ortiz, A., Vidal-Silva, C.: Factores asociados al éxito de los estudiantes en modalidad de aprendizaje en línea: Un análisis en minería de datos. Formación Universitaria, vol. 13, no. 6 (2020)
16. Marco, V., Jarauta, E.: Burden of disease calculation of cardiovascular risk and therapeutic objectives. Clínica e Investigación en Arteriosclerosis, vol. 33, pp. 10–17 (2021)
17. Millan-Solarte J. C., Cerezo, E. C.: Modelos para otorgamiento y seguimiento en la gestión del riesgo de crédito. Mét. Cuantitativos para la Eco. y la Emp., vol. 25, pp. 23–41 (2018)
18. Jalomo, J., Preciado, E.: Comparativa de desempeño de los optimizadores Adam vs SGD en el entrenamiento de redes neuronales convolucionales para la clasificación de imágenes ECG. Rev. Pistas Educativas, vol.42, pp. 313–325 (2020)
19. Ortíz, J. M. Rúa-Vieites, A., Bilbao-Calabuig, M. P.: Aplicación de árboles de clasificación a la detección precoz de abandono en los estudios universitarios de administración y dirección de empresas. Rev. Recta, vol. 18, pp. 177–201 (2017)
20. Gracia, M. E.: Máquinas de soporte vectorial y árboles de clasificación para la detección de operaciones sospechosas de lavado de activos. Lámpsakos, vol. 21, pp. 26–38 (2019)
21. Cobo, M. G.: Determinantes de malnutrición en pacientes en hemodiálisis: Efecto de la suplementación Proteica oral intradiálisis. Ph. D. Tesis, Departamento de Medicina, Universidad Complutense de Madrid, España (2018)

ISSN 1870-4069

RNN-LSTM Applied in a Temperature Prediction Model for Greenhouses

Juan M. Esparza-Gómez[1, 2], Héctor A. Guerrero-Osuna[2], Gerardo Ornelas-Vargas[2], Luis F. Luque-Vega[3]

[1] Universidad Autónoma de Zacatecas,
Posgrado en Ingeniería y Tecnología Aplicada,
Zacatecas, Mexico

[2] Universidad Politécnica del Sur de Zacatecas,
Zacatecas, Mexico

[3] Centro de Investigación, Innovación y Desarrollo Tecnológico,
Universidad del Valle de México,
Jalisco, Mexico

juan.esparza@upsz.edu.mx

Abstract. The climatic variables in protected agriculture are essential factors for the plant growth and, in general, of the entire crop. A good forecast of variables such as the internal temperature could help farmers to prevent losses in the harvest. In the present paper, the temperature forecast inside a greenhouse is obtained by implementing Deep Learning tools. The topology used for the temperature forecast was Recurrent Neural Networks (RNN) with Long-Short Term Memory (LSTM) algorithm. It is a type of neural network known to be suitable for processing time-series data. The analysis is performed with the many to one configuration for three input elements and one output element. The metrics used for the data analysis and validation (RMSE, MAE, R^2, and C_{eff}); it was observed that they significantly improve when the internal temperature is incorporated as part of the input elements in the combinations for forecasting. The results obtained with the RNN-LSTM provide RMSE values less than 0.30 and R^2 greater than 0.90, with a forecast interval of one hour into the future for Internal Temperature. It is shown that the LSTM algorithm within the RNN is an effective tool for a good forecast in time series, significantly helping the forecast of climatic variables in protected agriculture.

Keywords: RNN-LSTM, temperature prediction, deep learning.

1 Introduction

A good of climatic variables is essential for outstanding management in agriculture, either in the open air or in greenhouses. Therefore, a correct action that helps the observation and reasonable interpretation of the variables is through monitoring [1].

In the fourth industrial revolution, digital technology is made from integrating data and the connection of resources, creating an efficient and sustainable.

It has a presence in the agricultural world by implementing Intelligent components, creating interconnection of systems and machines. Its main objective is to have better production systems through the adaptability of monitoring and control systems, increasing the efficiency of production systems, fundamentally by optimizing energy consumption, machinery automatic control, water usage, fertilizers, and phytosanitary products, giving rise to what has been called Precision Agriculture [29]. Agriculture 4.0 uses technologies such as the Internet of Things, Big Data, Artificial Intelligence, Embedded Systems, Cloud Computing, Remote Sensing, among others, part of Industry 4.0.

Applications of these technologies can significantly improve the efficiency of agricultural activities [24, 25, 26]. Low-cost sensors and actuators can now connect to network platforms and upload their data to a remote database where Big Data analysis can occur. These network platforms aim to optimize the production efficiency, increasing quality, minimizing environmental impacts, and reducing the use of resources such as energy, water, and other consumables [27, 28] conducted a survey on the application of Big Data to agriculture.

They have pointed out that Big Data is now used to provide farmers with predictive insights in agricultural operations and operational decisions in real-time from monitoring by implementing artificial intelligence as a prediction system. Miranda et al. [4] mention the great importance of carrying out effective monitoring of the climatological variables of interest. In the present investigation, monitoring is carried out considering the variables of most significant interest for the study to make an efficient prediction of the internal temperature, to anticipate the appearance of possible problems in the culture [5].

The monitoring carried out in this research is carried out using advanced computational tools and since, in present investigations, strategies have been carried out through Artificial Intelligence (AI) to detect levels of interest in the behavior of the variables, guaranteeing the control and efficiency in crop productivity. [6]. Recurrent Neural Networks (RNN) are one of the AI topologies that have been used in recent years with acceptable results for the prediction of time series. RNN functions accurately as an identifier of trends for data and patterns are suitable for forecasting applications such as Deep Learning (DL) [2].

Jha et al. [8] show the RNN supplemented with the Long-Short Term Memory (LSTM) algorithm as predictors with levels of reliable precision if fed with a significant set of variables of interest for the prediction of time series in forecasting models for greenhouses. Jung et. al [9] as well as Hongkang et. al [10] propose models based on RNN with LSTM (RNN-LSTM) algorithms [11]. Gharghory [11] uses metrics such as the Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of variance to evaluate the prediction precision of an RNN-LSTM model and compare it to other models [3].

In this study, an approach to forecasting the internal temperature of a greenhouse is developed using external and internal climate data captured for a given period by a weather station and sensors connected to it.

Different RNN-LSTM structures configured through hyperparameters of interest were trained and tested with collected data. All RNN-LSTM were evaluated with metrics suggested in [11] and compared with different forecast approaches to assess the

goodness and suitability of this approach for the greenhouse internal temperature forecasting.

The LSTM algorithm has a significant advantage in expanding the memory capacity of the neural network. This characteristic leads to keeping a vast set of background data as a reference for the forecasting system. Keeping a large amount of data can significantly impact the accuracy of the prediction, reducing the RMSE and MAE to 0.5 and 0.004, respectively [2].

2 Related Works

Numerous investigations have been carried out with the objective of forecasting temperature, humidity, solar radiation, and other variables within protected environments such as greenhouses. All these are to determine the growth behavior of the crop [9]. There are several forecasting models.

However, in recent years' predictors based on Artificial Neural Networks have gained importance due to the range of tools provided by Machine Learning and the structures of algorithms. Dae-Hyun et al. [9] show comparisons between different structures considering various learning algorithms for the time series prediction. Abdulkarim et al. [11] show the advantage of the RNN, which can feedback the neuron output signal to the same neuron in the next time step.

The metrics usually used to assess LSTM prediction performance are the mean square error (MSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE), the square root of the mean square error (RMSE), and the Nash-Sutcliffe coefficient of efficiency (NSCE) [9, 11]. The LSTM algorithm has a significant advantage in expanding the memory capacity of the neural network. This characteristic leads to keeping a vast set of background data as a reference for the forecasting system. Keeping a large amount of data can significantly impact the accuracy of the prediction, reducing the RMSE and MAE to 0.5 and 0.004, respectively [2]. Singh [16] implements the RNN-LSTM to work with time series to forecast the Temperature and Relative Humidity inside a greenhouse.

For the temperature model, the metrics implemented for its validation were the MAE, RMSE, and R^2, obtaining MAE values of 0.488 for the temperature forecast, guaranteeing that the reliability of the forecast is within ±1°C. The RMSE obtained is 0.865, and the coefficient of determination R^2 is 0.953, which indicates that the general dispersion is small and does not cause a significant error with the observed temperature.

The RNN-LSTM training datasets can be selected in two ways. One way can be with 90% of the data sequence and the remaining 10% for testing and validation of the network. The other way is with 80% of the data sequence for training and the remaining 20% for network testing and validation. All the data must be normalized [3].

3 Methodology

The present research work presents new contributions for forecasting the green-house internal temperature using RNN and Deep Learning, using the Long Short-Term

Table 1. Climate Variables considered for this study.

Nomenclature	Climate Variable	Units
Ti	Internal Temperature	°C
To	External Temperature	°C
Ho	External Humidity	%
Hi	Internal Humidity	%
Di	Dew Point	%
Rs	Solar Radiation	W/m²

Fig. 1. Davis Vantage Pro 2 Weather Station.

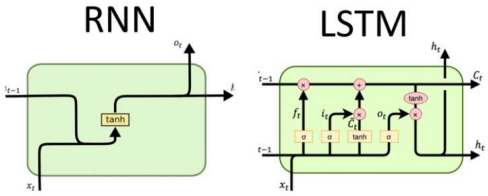


Fig 2. Data flow at time step t, Copyright 2020 by MathWorks Inc.

Memory (LSTM) algorithm as the one presented by Hochreiter and Schmidhuber in 1997 [17].

The data collection of the climatic variables was carried out inside (Internal Temperature and Relative Humidity) and outside (External Relative Humidity, Solar Radiation, Outdoor Temperature, Wind Direction, and Wind Speed) a greenhouse (Table 1), using a Davis Vantage Pro2 meteorological station (Fig.1).

The greenhouse has a curved roof (165m2in area, 27.5 m long, 6 m wide) and plastic cover. It is traditionally used without any climate control equipment inside and relies only on natural ventilation.

The greenhouse is in the Mezquitera Sur, Juchipila, Zacatecas, Mexico. The data collection was carried out from July 12, 2020, to November 6, 2020, with a 5-minute sampling time for all the climatic variables of interest. A total of 33,696 samples were recorded for training and testing of the RNN-LSTM.

The RNN-LSTM topology is based on a generalization of the feedforward neural network that has internal memory. RNN is recurrent at the neuron level as the neuron output is calculated using the current inputs and its previous output. The RNN considers the current inputs and the output that it has learned from the before deciding.

Table 2. Components of the data flow over time t.

Nomenclature	Definition	Formula
i_t	Input Gate	$i_t = \sigma(W_i x_t + R_i h_{t-1} + b_i)$
f_t	Forgate Gate	$f_t = \sigma(W_f x_t + R_f h_{t-1} + b_f)$
C_t	Cell candidate	$C_t = \tanh(W_g x_t + R_g h_{t-1} + b_g)$
O_t	Ouput Gate	$o_t = \sigma(W_o x_t + R_o h_{t-1} + b_o)$

Table 3. Hyperparameter settings sets.

Parameter	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6	Value 7
Input Size	3	3	3	3	3	3	3
Number of Rsponses	1	1	1	1	1	1	1
Hidden Units	2	2	1	1	1	1	1
Number of Epochs	200	250	300	300	300	350	300
Mini Batch Size	720	650	620	620	620	800	700
Gradient Threshold	0.9	0.5	0.6	0.8	0.9	0.6	0.7
Learning Rate	0.02	0.1	0.001	0.005	0.001	0.002	0.005
Learn Rate Schedule	150	200	200	125	125	120	200
Learn Rate drop Factor	0.5	0.2	0.2	0.2	0.2	0.3	0.5
Number of hiden units	100	200	250	250	300	350	400

The ADAM algorithm was adopted to make the calculation of the LSTM network more efficient. The architecture of the RNN-LSTM is observed in Fig. 2. The behavior of the data flow over time t is shown in Table 2.
where **σ** is the sigmoid activation function for the entry, forgetting, and exit gates respectively, the activation function **tanh** for the candidate gate, **W** the vector of weights for the entry, forgetting, and exit gates respectively, the function activation **tanh** for the candidate gate, **R** is the vector of recurring weights for the entry, forgetting, candidate and exit gates, **ht-1** is the output of the previous cell and **b** is the bias vector for the entry gates, oblivion, candidate and exit respectively. The most important things to consider when training and testing Neural Networks containing the LSTM algorithm are hyperparameters. These can affect its precision and performance [18, 19, 9, 20].

In Table 3, all the hyperparameters to be considered are shown. For this type of network, the most important are:

- Learning rate,
- Number of units of the hidden layer and,
- Mini Batch Size [14].

The number of units in the hidden layer will influence the adjustment effect. If the Mini batch size is too small, the training data will be challenging to converge, leading to a mismatch. Many consulted papers start with a high learning rate and lower it as the training goes on. We noted that the learning rate is very dependent on the network architecture. If the learning rate is too large, the required memory will increase

Table 4. Metrics for the evaluation of the efficiency of the RNN-LSTM.

Evaluation	RMSE	C_{eff}
Very Good	≤ 0.30	≥ 0.91
Good	0.30-0.40	0.84 - 0.91
Acceptable	0.40-0.50	0.75 - 0.84
No Acceptable	> 0.50	< 0.75

significantly [11]. The experimental environment consisted of an Intel (R) Core (TM) i5-9300H2.40 GHz quad-core processor with a 16 GB memory. The operating system was Windows 10 64-bit; the programming was carried out in MATLAB software.

The various structures were obtained from the hyperparameters variation. Table 3 shows the different hyperparameter settings used for this work. The criteria for evaluating the goodness and suitability of the network based on the fit are shown in Table 4 [21]. The MAE is used to reflect prediction errors, and its range is [0, +∞). When the predicted and observed values are identical, the MAE is zero, indicating a perfect model. Significant errors lead to high MAE values [22].

Other metrics to consider in the analysis to check the goodness of the network are the coefficient of determination (R^2) and the Coefficient of efficiency (C_{eff}) that help determine the closeness of the predicted data with the observed data. Both coefficients range is [0, 1], [9, 21]. The number of combinations made was obtained from five variables of interest; the internal temperature was excluded, in arrays of 3 input elements through formula 1:

$$C_r^n = \frac{n!}{(n-r)!\,r!}, \tag{1}$$

where:

n = Total climatic variables considered.

r = Number of variables considered for each arrangement.

From this, a total of 10 combinations were obtained to perform the tests with the RNN-LSTM. The metrics analyzed were the Mean Square Error (RMSE), the Absolute Mean Error (MAE) [3], and the Determination Coefficient (R^2) [23].

Later, seeking to improve the prediction results for the internal temperature (T_i), the five inputs combinations with best RMSE and C_{eff} were selected, and one input variable in each combination was substituted with the internal temperature, generating new input combinations.

The RNN-LSTM was trained and tested with the 80-20 arrangement, 80%training data, and 20% test data. The metrics analyzed were the Mean Square Error (RMSE), the Absolute Mean Error (MAE) [3], and the Determination Coefficient (R^2) [23].

4 Results

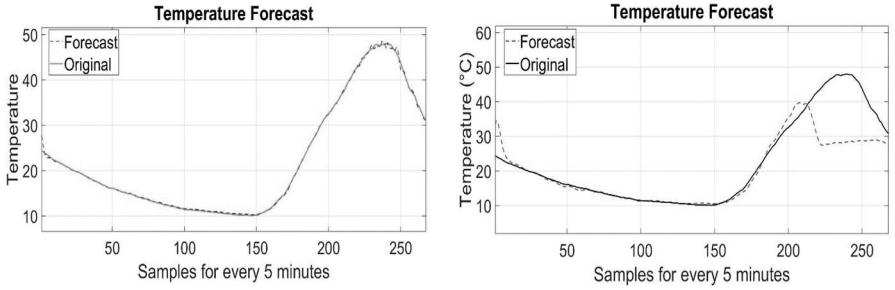
The best results for the RNN-LSTM training and testing were achieved with the hyperparameters set shown in column Value 4 of Table 3. These results were validated using the metrics shown in Table 5 and Table 6. The RMSE values shown in Table 5 have a range from 0.30027 to 5.5629.Fig. 3a, shows the forecast vs. actual value of an

Table 5. Sequence of input-output variables (Many to one).

put Secuence	RMSE	MAE	R^2	C_{ef}
Hi-Id-Rs	0.3003	0.0095	0.9994	0.9
Hi-Id-To	0.3128	0.0123	0.9993	0.9
Hi-Id-Ho	0.3966	0.0099	0.9989	0.9
Id-Rs-Ho	1.4368	0.0471	0.9857	0.9
Hi-Ho-To	1.8531	0.0451	0.9762	0.9
Id-Ho-To	3.3477	0.0721	0.9223	0.9
Hi-To-Rs	3.4352	0.0783	0.9182	0.9
Id-Rs-To	4.2746	0.0730	0.8734	0.8
Ho-To-Rs	5.3143	0.0853	0.8043	0.8
Hi-Rs-Ho	5.5629	0.3228	0.7856	0.7

Table 6. Comparison of data obtained in the different training sequences (Many to one).

Input Secuence	RMSE	MAE	R^2	C_{eff}
Hi-Id-Ti	0.1493	0.00407	0.9998	0.9998
Ho-To-Ti	0.1876	0.0077	0.9998	0.9998
Hi-Ti-To	0.2081	0.0075	0.9997	0.9997
Hi-Ho-Ti	0.2118	0.0059	0.9997	0.9997
Id-Rs-Ti	0.2507	0.0127	0.9996	0.9996



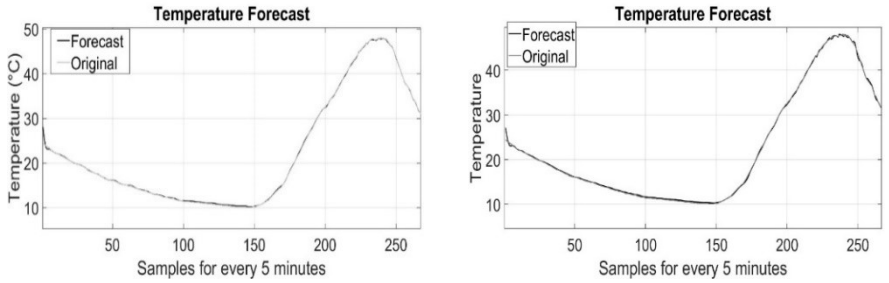
(a) (b)

Fig. 3. Behavior curves of the temperature forecast first sequence of input variables.

RNN-LSTM with 0.3 RMSE, while Fig. 3b, shows the forecast vs. actual value of an RNN-LSTM with 5.5629RMSE. As for the MAE values from 0.0095 to 0.32. For theR2coefficient, the range of values was 0.7856 to 0.9994, and C_{eff} values from 0.9994 to 0.7970. Subsequently, a new training pattern for the RNN-LSTM using the internal temperature (T_i) within the input sequence as shown in the following Table 6 was tried out. The results obtained had RMSE values from 0.14931 to 0.2507, MAE values from 0.00407 to 0.0127, and R^2up to 0.9998, and C_{eff} under to 0.2507, which presents a solid test forecast relationship.

Table 7. Comparison of parameters obtained.

Implemented model	RMSE	MAE	R^2	C_{eff}
Hi-Id-Ti	0.1493	0.00407	0.9998	0.9998
Ho-To-Ti	0.1876	0.0077	0.9998	0.9998
Hi-Ti-To	0.2081	0.0075	0.9997	0.9997
Hi-Ho-Ti	0.2118	0.0059	0.9997	0.9997
Id-Rs-Ti	0.2507	0.0127	0.9996	0.9996
[32] (RNN)	1.7963	1.3431	–	–
[32] (LSTM)	1.8044	1.3521	–	–
[32] (EEMD-LSTM)	0.7098	0.5336	–	–
[33] (RNN)	0.865	0.488	0.953	–
[31] (MLP-BPP)	0.711	0.558	0.980	–
[30] (CFD)	2.3518	2.0312	–	–



(a) (b)

Fig. 4. Behavior curves of the temperature forecast, second sequence of input variables.

The forecast graphs (Fig. 4a and Fig. 4b) show the similarities and slight differences between forecast and actual values. The prediction model makes forecasts very close to the observed values. From the metrics obtained, it was observed that the five sequences that contain T_i yielded good forecast results of the greenhouse internal temperature. In Table 7 a comparison of statistical metrics between this work and others found in the literature for the same application is shown.

5 Conclusions

The obtained results show that the combination of RNN-LSTM algorithms and a good selection of input variables can yield outstanding forecasting results. Metrics values such as RMSE = 0.1493 and C_{eff} = 0.9998 were obtained, which are considered by the literature as very good for the behavior of the RNN with LSTM algorithm. It is also observed that the predicted values and the observed values are very close.

This is corroborated by the values of the coefficient of determination R^2 with results of 0.9998, and the MAE = 0.00407. These metrics values were achieved thanks to the

relationship that exists between the internal dew (Id), internal relative humidity (Hi), and the internal temperature (Ti)variables in the greenhouse.

The applied model of RNN-LSTM has proven to be a tool that generates a clear interpretation of the training data. With a somewhat limited amount of data of 33,696, 80% of the data was enough to provide a model with excellent metrics when forecasting the greenhouse's internal temperature.

Future work is considered to make other prediction models using alternative ANN topologies such as Convolutional Neural Networks with Long-Short Term Memory algorithm (CNN-LSTM) and Support Vector Regression (SVR). This work will allow a fair comparison for these topologies when they are applied to this application.

References

1. Rodríguez, S., Gualotuña, T., Grilo, G.: A system for the monitoring and predicting of data in precision agriculture in a rose greenhouse based on wireless sensor networks. Procedia Computer Science, vol. 121, pp. 306–313 (2017) doi: 10.1016/ j.procs.2017.11.042
2. Poornima, S., Pushpalatha, M.: Prediction of rainfall using intensified lstm based recurrent neural network with weighted linear units. Atmosphere, vol. 10, no. 11, pp. 1–18 (2019) doi: 10.3390/atmos10110668
3. Xie, A., Yang, H., Chen, J., Sheng, L., Zhang, Q.: A short-term wind speed forecasting model based on a multi-variable long short-term memory network. Atmosphere, vol. 12, no. 5, pp. 1–17 (2021) doi: 10.3390/atmos12050651
4. Castañeda-Miranda, A., Icaza-Herrera, M., Castaño, V. M.: Meteorological temperature and humidity prediction from fourier-statistical analysis of hourly data. Advances in Meteorology, vol. 2019, pp. 1–13 (2019) doi: 10.1155/ 2019/4164097
5. Li-Li, G., Yong, L., Zi-Wei, L., Hua-Wei, S.: Application of particle swarm optimization bp algorithm in air humidity of greenhouse crops. In: IOP Conference Series: Materials Science and Engineering, vol. 565 (2019)
6. Izzatdin, A., Ismail, M. J., Haron, N., Mehat, M.: Remote monitoring using sensor in greenhouse agriculture. In: 2008 International Symposium on Information Technology, pp. 1–8 (2008) doi: 10.1109/ITSIM.2008.4631923
7. Lu, Y. S., Lai, K. Y.: Deep learning-based power generation forecasting of thermal energy conversion. Entropy, vol. 22, no. 10, pp. 1–15 (2020) doi: 10.3390 /e22101161
8. Jha, K., Doshi, A., Patel, P., Manan, S.: A comprehensive review on automation in agriculture using artificial intelligence. Artificial Intelligence in Agriculture, vol. 2, pp. 1–12 (2019) doi: 10.1016/j.aiia.2019.05.004
9. Jung, D. H., Kim, H. S., Jhin, C., Kim, H. J., Park, S. H.: Time-serial analysis of deep neural network models for prediction of climatic conditions inside a greenhouse. Computers and Electronics in Agriculture, vol. 173, pp. 2–11 (2020) doi: 10.1016/j.compag.2020.105402
10. Hongkang, W., Li, L., Yong, W., Fanjia, M., Haihua, W., Sigrimis, N. A.: Recurrent neural network model for prediction of microclimate in solar greenhouse. IFAC-Papers Online, vol. 51, no. 17, pp. 790–795 (2018) doi: 10.1016/j.ifacol.2 018.08.099
11. Abdulkarim, S. A., Engelbrecht, A. P.: Time series forecasting using neural networks: Are recurrent connections necessary? neural processing letters, vol. 50, pp. 2763–2795 (2019) doi: 10.1007/s11063-019-10061-5
12. Wang, K., Qi, X., Lui, H.: Deep belief network-based k-means cluster approach for short-term wind power forecasting. Energy, vol. 165, pp. 840–852 (2018) doi: 10.1016/j.energy.2018.09.118
13. Zhang, X., Zhang, G., Nie, Z., Gui, Z., Que, H.: A novel hybrid data-driven model for daily land surface temperature forecasting using long short-term memory neural network based

on ensemble empirical mode decomposition. Int. J. Environ. Res. Public Health, vol. 15, no. 5, pp. 1032 (2018) doi: 10.3390/ijerph15051032
14. Karevan, Z., Suykens, J.: Transductive LSTM for time-series prediction: An application to weather forecasting. Neural Networks, vol. 125, pp. 1–9 (2020) doi: 10.1016/j.neunet.2019.12.030
15. Tran, T. T. K., Bateni, S. M., Ki, S. J., Vosoughifar, H.: A review of neural networks for air temperature forecasting. Water, vol. 13, no. 9, pp. 1294 (2021) doi: 10.3390/w13091294
16. Singh, V. K., Tiwati, K. N.: Prediction of greenhouse microclimate using artificial neural network. Applied Ecology and Environmental Research, vol. 15, no. 1, pp. 767–778 (2017) doi: 10.15666/aeer/1501_767778
17. Nacimento, S., Valdenegro-Toro, M.: Modeling and soft-fault diagnosis of under-water thrusters with recurrent neural networks. IFAC-PapersOnLine, vol. 51, no. 29, pp. 80–85 (2018) doi: 10.1016/j.ifacol.2018.09.473
18. Outanoute, M., Lachhab, A., Ed-Dahhak, A., Selmani, A., Guerbaoui, M., Bouch-ikhi B.: A neural network dynamic model for temperature and relative humidity control under greenhouse. In: 2015 Third International Workshop on RFID and Adaptive Wireless Sensor Networks (RAWSN), pp. 6–11 (2015) doi: 10.1109/ RAWSN.2015.7173270
19. Ali, A., Hassanein, H. S.: Wireless sensor network and deep learning for prediction greenhouse environments. In: International Conference on Smart Applications, Communications and Networking (SmartNets), pp. 1–5 (2019) doi: 10.1109/Smart Nets48225.2019.9069766
20. Geng, D., Zhang, H., Wu, H.: Short-term wind speed prediction based on principal component analysis and LSTM. Appl. Sci. 2020, vol. 10, no. 13, pp. 4416 (2020) doi: 10.3390/app10134416
21. Agana, N. A., Homaifar, A.: EMD-Based predictive deep belief network for time series prediction: An application to drought forecasting. Hydrology, vol. 5, no. 1, pp. 18 (2018)
22. Pérez, I., Godoy, A.: Neural networks-based models for greenhouse climate control. In: Actas de las XXXIX Jornadas de Automática, pp. 875–879 (2018) doi: 10.3390/hydrology5010018
23. Hua, Y., Guo, J., Zhao, H.: Deep belief networks and deep learning. In: Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things. ICIT 2015, pp. 1–4 (2015). doi: 10.1109/ICAIOT.20 15.7111524
24. Klerkx, L., Jakku, E., Labarthe, P.: A review of social science on digital agriculture, smart farming and agriculture 4.0: New contributions and a future research agenda. NJAS - Wageningen Journal of Life Sciences, vol. 90-91, pp. 1573–5214 (2019) doi: 10.1016/j.njas.2019.100315
25. Rose, D. C., Chilvers, J.: Agriculture 4.0: Broadening responsible innovation in an era of smart farming. Front. Sustain. Food Syst., pp. 1–7 (2018) doi: 10.3389/fsufs.2018.00087
26. Lui, Y., Ma, X., Shu, L., Hancke, G. P., Abu-Mahfouz, A. M.: From industry 4.0 to agriculture 4.0: current status. Enabling Technologies, and Research Challenges, vol. 17, no. 6, pp. 4322–4334 (2020) doi: 10.1109/TII.2020.3003910
27. Zhai, Z., Martínez, J. F., Beltrán, V., Martínez, N. L.: Decision support systems for agriculture 4.0: Survey and challenges. Computers and Electronics in Agriculture, vol. 170, pp. 1–16 (2020) doi: 10.1016/j.compag.2020.105256
28. Raj, M., Gupta, S., Chamola, V., Elhence, A., Garg, T., Atiquzzaman, M., Niyato, D.: A survey on the role of internet of things for adopting and promoting agriculture 4.0. Journal of Network and Computer Applications, vol. 187, 1–29 (2021) doi: 0.1016/j.jnca.2021.103107
29. Pspocsi, L., Csoto, M.: Digital excellence in agriculture in Europe and Central Asia, good practices in the field of digital agriculture stocktaking report. ITUand FAO, pp. 1–196 (2021)

30. Xue, C., Dongming, L., Limin, S., Zhenhui, R.: A virtual sensor simulation system of a flower greenhouse coupled with a new temperature microclimate model using three-dimensional CFD. Computers and Electronics in Agriculture, pp. 1–14 (2021)
31. Singh, V. K., Tiwari, K. N.: Prediction of greenhouse micro-climate using artificial neural network, applied ecology and environmental research, pp. 767–778 (2017)
32. Xike, Z., Qiuwen, Z., Gui, Z., Zhiping, N., Zifan, G., Huafei, Q.: A novel hybrid data-driven model for daily land surface temperature forecasting using long short-term memory neural network based on ensemble empirical mode decomposition. International Journal of Environmental Research and Public Health, pp. 1–23 (2018)
33. Hongkang, W., Li, L., Yong, W., Fanjia, M., Haihua, W., Sigrimis, N.: Recurrent neural network model for prediction of microclimate in solar greenhouse. Science Direct, pp. 790–795 (2018)

ISSN 1870-4069

Optimization of Combinatorial Software Testing: A Systematic Literature Review

Puxka Acosta-Domínguez, Angel J. Sánchez-García,
Candy Obdulia Sosa-Jiménez

Universidad Veracruzana,
Facultad de Estadística e Informática,
Mexico

puxka.acodom@gmail.com, {angesanchez, cansosa}@uv.mx

Abstract. One of the reasons why Industry 4.0 is a reality is the development of quality Software that allows the interconnectivity and management of the devices we use. To ensure quality in the software, it is necessary that it be released with the least number of failures. For this, it is necessary that the software tests cover all possible paths extensively and it is for this reason that approaches such as combinatorial tests arise. This type of test seeks to generate test cases for the detection of failures. This Systematic Literature Review (SLR) compiles the optimization techniques, approaches, areas of opportunity and application of combinatorial tests. In this SLR, 82 primary studies were identified, based on the Kitchenham and Charters guide, where it is observed that the most frequent application of this type of test is the web, although most of the studies only propose improvements without any specific domain. Also, the most widely used approaches are IPOG and its variants. Finally, the most mentioned opportunity area was the improvement of results and algorithms.

Palabras clave: Combinatorial testing, optimization, systematic literature review, quality software.

1 Introduction

In recent years, the need for high-quality software has increased. This is due to the fact that all current devices and those generated in Industry 4.0 require Software that is less prone to failures, and that quality does not impact human and economic resources. This quality is ensured when the Software is thoroughly tested before it is released.

The activities carried out in the development of a software project are divided into processes, such as requirements, design, construction, testing, configuration management, among others [1], being the testing stage the interest of this document. Since depending on the system and the defect sought, it is the type of testing technique that will be used [2]. With the interest in combinatorial tests, this paper presents a Systematic Literature Review (SLR) focused on the categorization of current research.

The aim is to find techniques to design combinatorial tests that can be used for future research areas. This work is aimed at researchers to find niches of opportunity in this area. Other beneficiaries will be students and developers, especially those in the testing

Table 1. Research questions.

Quesition	Motivation
Q1.- What are the applications of combinatorial test?	To know where the combinatorial tests have currently been applied.
Q2.- Which optimization techniques have been proposed to generate combinatorial test cases?	To identify proposals for the Artificial Intelligence optimization area in the generation of combinatorial test cases.
Q3.- What areas of opportunity are there for the research community?	To present proposals for areas of research opportunity on the subject of combinatorial test design.

Table 2. Identified keywords and synonyms.

Concept	Synonyms
Combinatorial testing	Combinatorial test, combinatorial interaction testing, pairwise testing
Automation	Automated, automatized
Optimization	Optimize, optimized
Algorithm	Approach, strategy, method

area. The first will get an insight into a topic that has been little covered, and developers in the industry will get a starting or improvement point for their current testing. The work will use a systematic method to extract and analyze the works on the topic.

The aim of this Systematic Literature Review is to present the state of the use of Artificial Intelligence in the design of combinatorial tests to ensure the quality of the software. This is to discuss the most promising approaches and, in the future, generate lines of research in the area.

This paper is organized as follows: Section II presents the background and related work of reviews on the subject. Section III exposes the method followed to carry out the SLR. Section IV describes the final results. Finally, Section V draws conclusions.

2 Background and Related Work

Combinatorial tests are a method that can detect defects that other types of tests cannot (for example, random tests or equivalence partition tests), since the defect can be found in the combination of parameters. In addition, they reduce the number of tests, therefore reducing costs [3]. These tests are of the black box type, therefore, to carry them out, test cases are created that receive inputs to a functionality and as a result provide an output. Combinatorial tests can be from two to T variables. Combinatorial tests have also been called as combinatorial tests in pairs, which are those of two variables, and tests of combinatorial interaction.

All combinations can be represented by a covering array (CA). The CA is a mathematical object in which all combinations of a parameter are covered at least once [4]. These can be created with the help of tools, which change depending on the algorithm on which they are based. There are different types of algorithms, among those based on Artificial Intelligence (AI) are optimization algorithms. AI has sought to reduce the human error that is always present in any system, therefore, it is not

Table 3. Inclusion Criteria.

IC	Description
1	Studies published between the years 2015 to 2021
2	Studies written in English
3	The title and / or abstract give indications that at least one research question will be answered
4	The full text answers at least one research question

Table 4. Exclusion Criteria.

EC	Description
1	It is a summary, workshop, opinion piece, presentation, book or technical report.
2	Do not have access to the full text
3	It is a duplicate research
4	The full text answers at least one research question

surprising that this discipline collaborates with Software Engineering (SE). According to a study [5], the use of AI algorithms in the SE testing phase has been investigated since 1970 and mostly in black box testing, of which combinatorial testing is a part.

The oldest review on the subject found dates from 2014 [6] and it took the topic of combinatorial testing as part of Software Product Line testing strategies. Other review [7], collected the algorithms and tools from 1991 to 2014, but is not strict in following a method.

The only Systematic Mapping found [8] has as main focus combinatorial tests with application in SPL. A Systematic Literature Review from 2017 [9] talks about combinatorial tests in general, mentioning tools and limitations, but it is the one that has its method with less detail of all the reviews.

The SLR found [10] describes the combinatorial tests that have handling of restrictions, which are also mentioned in the present work, but not limited to these.

3 Research Method

To carry out this review, the method proposed in [11] by Kitchenham and Charters was performed, which is a guide used in Software Engineering to carry out Systematic Literature Reviews (SLR). This section describes the steps defined in [11].

3.1 Research Questions

Table 1 shows the research questions with their motivation, to guide this process.

3.2 Search Strategy and Data Sources

The automated search strategy was used to collect the information. For this, keywords were identified, which were taken from the research questions and similar terms, as shown in Table 2.

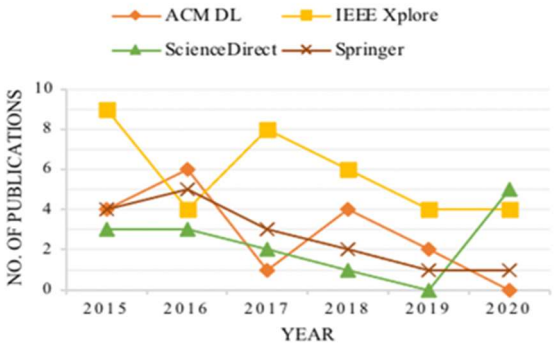


Fig. 1. Primary studies by year of publication.

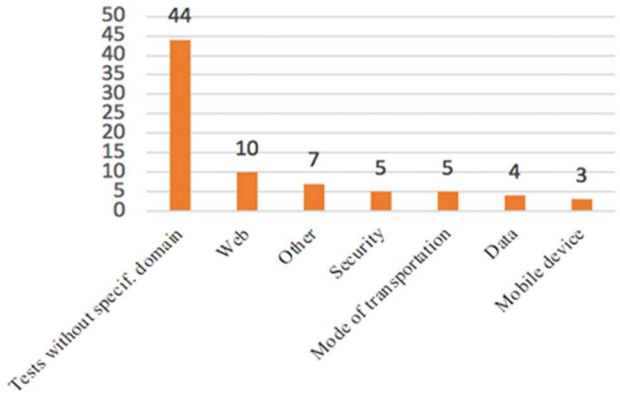


Fig. 2. Number of primary studies per application of combinatorial tests.

With these terms, 5 search strings were created and they were validated with the precision and recall criteria, which have been used in Systematic Reviews of Software Engineering [12]. The results of these metrics for each of the 5 search strings can be found in Appendix A[1].

The selection of sources was based on [11] and [12], two guides for conducting systematic reviews of Software Engineering. Following these, the following repositories were selected: SpringerLink, IEEE Xplore, ScienceDirect and ACM Digital Library.

3.3 Selection Criteria

For the selection of primary studies, inclusion and exclusion criteria were established, which can be seen in Table 3 and 4.

[1] Appendix A. Search String metrics: https://drive.google.com/file/d/1B_7ZuB0EFYNzbYmtQwAZt1SWhGk0ZA9R/view?usp=sharing

Table 5. Application of inclusion and exclusion criteria by stage.

Database	First Results	Stage 1	Stage 2	Stage 3
ACM Digital Library	400	228	30	17
IEEE Xplore	360	177	44	35
ScienceDirect	420	183	15	14
SpringerLink	433	168	22	16
Total	1613	756	111	82

Table 6. Quality assessment criteria.

NC	Criteria
1	Is the study research-based (not just a report based on "lessons learned" or opinion)?
2	Is there an explanation of the objectives of the research?
3	Is there a description of the context in which the research was carried out?
4	Does the research design speak to the research objectives?
5	Did the sample of data used have a description?
6	Was the information collected in a way that spoke to the research problem?
7	Was the information analysis rigorous?
8	Is there an explanation for the findings?
9	Is the study of value for research or practice?

3.4 Selection Procedure

The selection process is made up of the following stages:

- Stage 1. Primary studies are filtered according to IC1 and IC2 and studies are removed according to EC1.
- Stage 2. The primary studies are filtered according to the IC3 and the primary studies are removed according to the IC2 and EC3.
- Stage 3. Primary studies are filtered according to IC4.

First, the automated search was executed in the four repositories, being 1613 the works found, as can be seen in Table 5. After that, the inclusion and exclusion criteria were applied stage by stage, reducing the studies in SpringerLink by 95.75%, in IEEE Xplore by 90.28%, in ScienceDirect by 96.67% and in ACM Digital Library by 96.08%.

3.5 Quality Assessment

The criteria to assess the quality of the primary studies were obtained from a systematic review used and accepted in the Software Engineering community [13]. The criteria are in the form of a questionary. The selected criteria are shown in Table 6. Each question is answered with "yes" or "no" (rated as 1 or 0 respectively). So that each study can

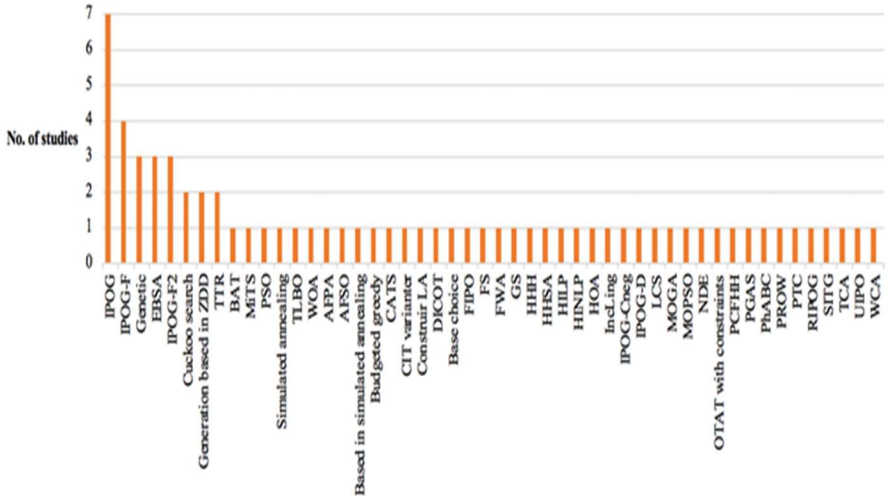


Fig. 3. Algorithms most used in combinatorial testing.

have a final score from 0 (which means very poor) to 9 (which means very good). The quality score of the selected works can be seen in a freely accessible spreadsheet [14].

4 Results

In Fig. 1 the primary studies that passed the selection process are shown. It was found that three digital libraries have been a decrease in the combinatorial test research. Each of the research questions is answered in summary below.

4.1 Q1: What are the Applications of Combinatorial Test?

As can be seen in Fig. 2, seven main applications of combinatorial tests were identified in the last 6 years. All studies whose approach to performing combinatorial tests did not have a specific application domain were pooled. In the *"Other"* category, the domains that were only present in a primary study were grouped.

4.2 Q2: Which Optimization Techniques Have Been Proposed to Generate Combinatorial Test Cases?

From all the studies, 54 papers answered this question, which represent 66% of the total number of studies selected. These mention around 50 optimization algorithms, all with a varied level of detail. Even with this variation, we organize them into the original algorithms, those that were based on an existing one to improve it as it is shown in Table 7 and those that use another algorithm as part of their own process, as can be seen

Table 7. Algorithms based in other optimization algorithms.

Base Algorithm	Proposed Algorithm
ABC	PhABC
AETG System	Prow
BSA	EBSA
Cuckoo Search	LCS
D-construction	RIPOG
FPA	AFPA
Genetic	GS, HOA, PGAS
Greedy approach	Budgeted greedy, DICOT, FIPO, HILP, HINLP, IncLing, IPOG, IPOG-F, IPOG-F2, OTAT, PROW, PTC, TTR
IPO	IPOG, IPOG-D, IPOG-F, IPOG-F2, UIPO
IPOG	RIPOG
IPOG-C	IPOG-CNeg
IPOG-D	RIPOG
OTAT	EBSA, FS, FWA, OTAT with constraint, PROW
PSO	MOPSO, SITG
Simulated annealing	AFSO
Tabu search	CATS

Table 8. Algorithms used with another optimization algorithm.

Base Algorithm	Proposed Algorithm
Ant Colony	UIPO
ExtendCA	AFSO
FPA	FIPO, PCFHH
Greedy Search	PCFHH

in Table 8. The detail of each algorithm per paper and its reference can be seen in Appendix B[2].

As can be seen in Fig. 3, the most used algorithm was In-Parameter-Order-General (IPOG) since ACTS, the most used tool, uses this algorithm and variations. The most used algorithm from the original algorithms is the Genetic Algorithm (GA), the second is Cuckoo search and the others were only used once.

Most of the original algorithms are bioinspired. Cuckoo search, Bat algorithm (BAT) and Whale Optimization Algorithm (WOA) are algorithms inspired by animal behaviors, Simulated Annealing in the process of heat treatment of metal, Teaching Learning Based Optimization (TLBO) in school teaching and Particle Swarm Optimization (PSO) in social behavior.

[2] Appendix B. Algorithms by paper: https://drive.google.com/file/d/19qJ8PsWntCOToq73-SIryjuMq-RF6PL3/view?usp=sharing

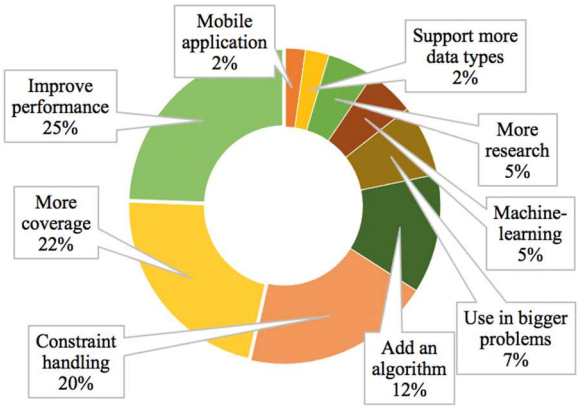


Fig. 4. Percentage of opportunity areas mentioned in the studies.

4.3 Q3.- What Areas of Opportunity are There for the Research Community?

The 9 areas of opportunity mentioned in the studies are shown in Fig. 4. One of the most mentioned was improving the performance of algorithms, although this type of test is known to give optimal results in its performance. It is wanted to decrease the size of the test set [15] or improve search efficiency in a large set of test cases [16]. The possibility exists of achieving this by parallelizing the algorithm [17]. Later, the majority mentioned the increase in coverage of variables. This in combinatorial tests in pairs [18], 3-variables [19] and one of 6-variables [20].

Another interest is adding constraint handling. Many of the tools of the last years do not implement any method for this [21]. Therefore, it is the intention to apply them in well-known algorithms [22] or in algorithms that have only just been introduced to combinatorial tests [23]. It also seeks to complement or combine an algorithm with some other to improve the results [24], but they want to incorporate more to refine the algorithm.

Also, testing the algorithm in big problems is mentioned. The goal would be to scale to very large problems and find smaller sets of tests than in your current results [25]. The sixth most mentioned is to integrate machine-learning to the algorithm. This could help test generation by learning from previously generated test sets [22].

5 Conclusions and Future Work

This work took an approach that few works have taken and, unlike those that have, the method proposed in [11] was detailed and followed. With this, the reproducibility of the results is possible. In addition, the results of the extraction and quality evaluation are online and available to all public. As a result, the algorithms that have been applied

to generate the combinations of test cases have been compiled. These algorithms are the basis for combinatorial tests, which are applied in a wide variety of fields such as web, security, mobile devices and others, which shows that it is an active area, current, but above all, a useful area to ensure the quality of software of any kind.

In this study, it was observed that the most widely used approaches are IPOG and its variants, bioinspired approaches and classic approaches such as Simulated Annealing and Greedy search. Therefore, it can be seen that although there is a growing interest in bioinspired algorithms, some other metaheuristics have contributed much knowledge in this area of Software Engineering.

Finally, the most mentioned opportunity area was the improvement of results and algorithms. This is important since based on the proposed approaches, you can explore and experiment with others to compare the results. Nevertheless, the greater coverage and handling of restrictions are also highlighted. Therefore, it is seen that there is still a lot of research to be done in this field and with the help of new optimization strategies, it is sought to ensure the quality of the software, which is fundamental in Industry 4.0.

As future work it is proposed to explore the comparison of bioinspired algorithms with other metaheuristics (such as simulated annealing). Also explore local search methods to get better coverage in test cases.

References

1. Bourque, P., Dupuis, R., Abran, A., Moore, J. W., Tripp, L.: Guide to the software engineering body of knowledge. IEEE Software, IEEE Computer Society, vol. 16, no. 6, pp. 35–44 (1999) doi: 10.1109/52.805471
2. Wu, H., Nie, C., Petke, J., Jia, Y., Harman, M.: An empirical comparison of combinatorial testing, random testing, and adaptive random testing. IEEE Trans. Softw. Eng., vol. 46, no. 3, pp. 302–320 (2020) doi: 10.1109/TSE.2018.2852744
3. Kuhn, R., Kacker, R., Lei, Y., Hunter, J.: Combinatorial software testing. Computer (Long. Beach. Calif), vol. 42, no. 8, pp. 94–96 (2009) doi: 10.1109/ MC.2009.253
4. Sloane, N. J. A.: Covering arrays and intersecting codes. J. Comb. Des., vol. 1, no. 1, pp. 51–63 (1993) doi: 10.1002/jcd.3180010106
5. Lima, R., Da Cruz, A. M. R., Ribeiro, J.: Artificial intelligence applied to software testing: A literature review. In: Iberian conference on information systems and technologies. CISTI, vol. 2020, no. June (2020). doi: 10.23919/CISTI49556. 2020.9141124
6. Machado, I. D. C., McGregor, J. D., Cavalcanti, Y. C., De Almeida, E. S.: On strategies for testing software product lines: A systematic literature review. Inf. Softw. Technol., vol. 56, no. 10, pp. 1183–1199 (2014) doi: 10.1016/j.infsof. 2014.04.002
7. Khalsa, S. K., Labiche, Y.: An orchestrated survey of available algorithms and tools for combinatorial testing. In: Proceedings International Symposium on Software Reliability Engineering, ISSRE, pp. 323–334 (2014) doi: 10.1109/ISS RE.2014.15
8. Lopez-Herrejon, R. E., Fischer, S., Ramler, R., Egyed, A.: A first systematic mapping study on combinatorial interaction testing for software product lines (2015) doi: 10.1109/ICSTW.2015.7107435
9. Abdullah, A., Hassan, R., Shah Z. A.: A Systematic literature review of combinatorial testing. Int. J. Advance Soft Compu. Appl, vol. 9, no. 2 (2017)
10. Ahmed, B. S., Zamli, K. Z., Afzal, W., Bures, M.: Constrained interaction testing: A systematic literature study. IEEE Access, vol. 5, pp. 25706–25730 (2017) doi: 10.1109/ACCESS.2017.2771562

11. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Department of Computer Science University of Durham, Durham, UK, Tech Rep (2007)
12. Dieste, O., Grimán, A., Juristo, N.: Developing search strategies for detecting relevant experiments. Empirical Software Engineering, vol. 14, no. 5, pp. 513–539 (2009) doi: 10.1007/s10664-008-9091-7
13. Dyba, T., Dingsøyr, T.: Empirical studies of agile software development: A systematic review. Information and software technology, vol. 50, no. 9-10, pp. 833–859 (2008) doi: 10.1016/j.infsof.2008.01.006
14. Acosta, P.: Estudios primarios. Estrategias para diseñar pruebas combinatorias (2020)
15. Balera, J. M., de Santiago, V. A.: A controlled experiment for combinatorial testing. In: SAST: Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing, pp. 1–10 (2016) doi: 10.1145/2993288.2993289
16. Wolde, B. G., Boltana A. S.: Combinatorial testing approach for cloud mobility service. In: AICCC 2019: Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference, pp. 6–13 (2019) doi: 10.1145/3375959.3375967
17. Al-Hajjaji, M., Krieter, S., Thüm, T., Lochau, M., Saake, G.: IncLing: Efficient product-line testing using incremental pairwise sampling. ACM SIGPLAN Not., vol. 52, no. 3, pp. 144–155 (2016) doi: 10.1145/2993236.2993253
18. Alazzawi, A. K., Rais, H. M., Basri, S., Alsariera, Y. A.: PhABC: A hybrid artificial bee colony strategy for pairwise test suite generation with constraints support. In: IEEE Student Conference on Research and Development, SCOReD´19. pp. 106–111 (2019) doi: 10.1109/SCORED.2019.8896324
19. Kampel, L., Simos, D. E.: Set-based algorithms for combinatorial test set generation. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9976, pp. 231–240 (2016)
20. Ahmed, B. S.: Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing. Engineering Science and Technology, an International Journal, vol. 19, no. 2, pp. 737–753 (2016) doi: 10.1016/j.jestch.2015.11.006
21. Petke, J.: Constraints: The future of combinatorial interaction testing. In: Proceedings - 8th International Workshop on Search-Based Software Testing, SBST´15, pp. 17–18 (2015) doi: 10.1109/SBST.2015.11
22. Li, N., Lei, Y., Khan, H. R., Liu, J., Guo, Y.: Applying combinatorial test data generation to big data applications. In: ASE 2016. Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering, pp. 637–647 (2016) doi: 10.1145/2970276.2970325
23. Ahmed, B. S.: Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing. Engineering Science and Technology, an International Journal, vol. 19, no. 2, pp. 737–753 (2016) doi: 10.1016/j.jestch.2015.11.006
24. Ericsson, S., Enoiu, E.: Combinatorial modeling and test case generation for industrial control software using ACTS. In: Proceedings IEEE 18th International Conference on Software Quality, Reliability, and Security, QRS´18. pp. 414–425 (2018) doi: 10.1109/QRS.2018.00055
25. Qi, R. Z., Wang, Z. J., Li, S. Y.: A parallel genetic algorithm based on spark for pairwise test suite generation. J. Comput. Sci. Technol., vol. 31, no. 2, pp. 417–427 (2016) doi: 10.1007/s11390-016-1635-5

ISSN 1870-4069

Machine Learning for the Retail Trade Behavior Analysis in Mexico

Patricia Soto-Vázquez, Guillermo Molero-Castillo,
Everardo Bárcenas, Rocío Aldeco-Pérez

Universidad Nacional Autónoma de México,
Facultad de Ingeniería,
Mexico

soto.holden@gmail.com, gmoleroca@fi-b.unam.mx,
{ebarcenas, raldeco}@unam.mx

Abstract. In recent years, machine-learning methods have gained prominence when used as a tool for data analysis in different areas such as the economy. This article presents the result of the data analysis of the retail trade in Mexico and the industrial branches that comprise it. For the data analysis, unsupervised learning was used, specifically clustering based on the K-means, through which it was possible to organize clusters with information on the characteristics of the different industrial branches of the retail trade. As a result, it was possible to identify that this sector, the source of many jobs, subsists in the midst of an aggressive, demanding market, with insufficient access to update its technology and complex administrative procedures.

Keywords: Machine learning, clustering, k-means, economics, retail trade.

1 Introduction

The use of information as a source of knowledge is not only limited to understand what the data represents, but also generates value in any area it is applied, which is essential in the context of the actual, globalized world. Thus, because of this constant and necessary search to make total utilization of knowledge, Artificial Intelligence, applied in Industry 4.0, benefits the society in the process of digital transformation, mainly driven by machine learning and deep learning [1].

Knowledge in economics is built on the analysis of information derived from the economic activity of a country [2]. From the categorization of economic activity, economic sectors emerge as pillars of growth and development [3] [4]. Therefore, within the economic context in which Mexico develops, a knowledge-based economy is necessary, with machine learning being an important pillar in the push towards the fourth industrial revolution, which has led to the application of data analysis algorithms in order to build an efficient and quality economic system.

In this sense, given this growing need for data analysis in the economic field, the use of machine learning, as a support tool for advanced data analytics, is important [5] [6], since it offers a wide variety of algorithms, among which are supervised, unsupervised, deep, reinforcement, and mixed, currently achieving an important position in response

to the extensive digitization and storage of data. On the other side, current machine learning is not only changing the way a product is produced, marketed and sold, but is also part of the study of economic growth, determined by the increase in productivity and income of a country [7, 8].

This study establishes the basis for analyzing economic development, measured based on improvements in the living conditions of the population. There is no doubt that the development of new products and companies, with unprecedented levels of automation and robotization, can transversally transform the economy and the labor market. Thus, to understand how trade is directly associated with people quality of life, it is worthwhile to focus efforts on the analysis of one of the economic sectors that have a predominant impact on the Mexican economy, this is, retail trade; which is the economic activity defined by the individual sale of goods and services directly to final consumers [9].

This activity (by its nature) is a component of the supply chain in view to its model focused on the sale between the company and the consumer. This type of trade is a fundamental sector in Mexico, since, in terms of gross domestic product (GDP), tertiary activities had an annual percentage structure of 60%, corresponding to 2020, within which 9.2% corresponds to trade retail [10, 11]. In addition to the above, misinformation on the demeanor of economic activity not only leads people and companies to mismanage their business but encourages disinterest in establishing measures or laws that benefit the retail trade [12].

Consequently, this paper aims to show insight about retail trade in Mexico, which represents a field of opportunity, through an analysis, based on machine learning, since, moreover to its considerable percentage share with respect to GDP, it also concentrates a large population that finds, in this sector, a source of employment. The document is organized as follows, Section 2 presents the antecedents of economics as social science, some of the contributions of machine learning in the economy, discuss its applications, the use of algorithms and related work are also presented. Section 3 describes the method established as a proposed solution. Section 4 presents the results obtained, based on an example of application, and Section 5 summarizes some conclusions and future work.

2 Background

2.1 Retail Trade

Retail trade is defined by economic units, within which are several establishments that are under the control of a proprietary entity, permanently established and delimited by fixed facilities [10, 13]. Furthermore, these economic units are located at different geographical levels. For example, country, state, municipality, and locality, where they perform the task of enabling activities of buying and selling merchandise, or providing services, regardless of whether they have mercantile purposes [6] [15]. This group includes micro-businesses and small and medium-sized enterprises (SMEs) [14]. At present, the SME sector is one of the most vulnerable since, like any business, they require correct management of their financial income.

These incomes depend, largely, on proper financial management, which many SMEs lack. According to the Development Center for Business Competitiveness, 75% of SMEs close their operations just two years after being created. Moreover, the National Institute of Geography and Statistics (INEGI, by its acronym in Spanish) denotes that the new businesses in Mexico only live on average 7.7 years [16].

Conventionally, the study of economic growth is based on the analysis of indicators such as GDP, thanks to which the significant share of retail trade in the Mexican economy is notable. Although GDP is not a sufficient indicator to determine the economic growth of the country, it is one of the most important, since a rise in this indicator could easily translate into an increase in employment. In addition, the occupation and employment indicators provide relevant information in this area.

2.2 Economic Indicators

Concerning the employed population by size in the economic unit, the employment and occupation indicators establish that 20.1 million people are employed in micro-businesses, 7.5 million in small businesses, and 5.2 million in medium sized businesses.

This information is representative since it has a coverage of 63.9% (84556) of the dwellings in the National Survey of Occupation and Employment (ENOE, by its acronym in Spanish) [17]. Among the organizations that provide information on the activities, economic indicators, and the labor market include:

- The National Survey of Occupation and Employment, which is the primary source of information on the labor market. It provides monthly and quarterly data on the labor force, occupation, labor informality, underemployment, and unemployment [18]. In 2020, they disseminated the occupational characteristics of the population aged 15 years and over, along with demographic and economic variables for the analysis of the labor force.
- The Annual Trade Survey (EAC, by its acronym in Spanish), which provides information on commercial activities and provides a frequent statistical overview that contributes to the decision-making of the different productive sectors of the country [19]. The Annual Trade Survey is based on the Monthly Survey on Commercial Companies (EMEC, by its acronym in Spanish), whose main purpose is to generate statistical information.
- The World Trade Organization (WTO) is the only international organization, of which Mexico is a member, which deals with the rules that govern trade between countries. Its aim is to ensure that commercial exchanges take place in a fluid, predictable and free manner [20]. One of his recent publications was 'Helping MSMES Navigate the Covid-19 Crisis', which explains how SMEs have been affected by the COVID-19 pandemic.

2.3 Clustering Based Machine Learning

Machine learning consists of a set of algorithms for data-driven analysis, which allow establishing models, from the data of examples or experiences, to train machines (computers) and learn from them [8] [21]. Within machine learning, unsupervised methods are algorithms that base their training process on a previously defined,

labelless data set. That is, no target or class value is known, either categorical or numeric. Therefore, these methods do not require human intervention [22].

The main applications of unsupervised learning are related to data clustering, where the objective is to find clusters with similar elements, in such a way that the internal elements of a cluster have a high similarity, and are different (dissimilar) with elements of others clusters [23]. There are two main types of clustering algorithms [24]:

1. Hierarchical, which produce a hierarchical organization of the elements that make up the data set, thus enabling different levels of clustering.
2. Partitional, which generate clusters of elements that do not correspond to any type of hierarchical organization. These algorithms are based on the distance between elements.

Partitional algorithms assume a priori knowledge of the number of clusters into which the data set must be divided, that is, they arrive at a division that optimizes a predefined criterion [25]. Among the algorithms that use this type of clustering highlights K-means, whose main idea is to define k centroids (one for each cluster) and then take each element (a record) from the database and place it in the nearest centroid cluster.

The centroid is a point that occupies the middle position in a cluster. The next step is to recalculate the centroid of each cluster and redistribute all elements according to the nearest centroid. The process is repeated until there are no longer changes in the clusters formed [24]. In addition, in K-means the elbow method is used, with different configurations of k, to obtain an approximation to the adequate number of clusters [26].

2.4 Related Work

In recent years, in Mexico, trade is the object of study due to its importance, not only in the national and international economy but also because of its impact on the Mexican population. In this sense, the analysis of retail trade is essential for the benefit of understanding the development and economic growth of a certain region over the years.

This type of analysis can be achieved based on the observation of similarities, trends, and behaviors, for which machine learning algorithms are useful, some works related to data grouping and partitioning algorithms in economics are:

– Data mining and machine learning in the retail business: developing efficiencies for better customer retention [6] presented an analysis of retail marketing and discussed the application of data mining and machine learning techniques. Within the methodology, the use of K-means as a tool for the identification of incomplete data points stands out.

  This algorithm, together with another one for predicting customer interest and pattern mining techniques, made it possible to identify purchase patterns from user records.
– P2V-MAP: Mapping of market structures for large retail assortments [15], where market structures were analyzed through advances in natural language processing and machine learning. The approach used made it possible to compare data dimensionality reduction techniques, which show a contribution to the market analysis. In addition, the use of machine learning algorithms is proposed to propose solutions in problems related to the structures of the retail market.

- Machine learning for enterprises: Applications, algorithm selection, and challenges [4], in this work the importance of machine learning applied to companies, is exposed, with the aim of promoting their technological development, and thus reducing costs of products and services. In addition, the use of methods and challenges in the application of machine learning algorithms for grouping, classification, and forecasting were analyzed. As well as the increase in the implementation of machine learning tools and algorithms in companies to increase their potential.
- Sustainability of SMEs in the Competition: A Systemic Review on Technological Challenges and SME Performance [27], in this paper the importance of SMEs as an engine of economic development was discussed, described the challenges they face, and reviewed the need for technological progress to drive innovation in the economy and the positive effects it has on production levels and economic growth. In addition, the adoption of information technologies as a means to face competitive challenges in SMEs was exposed.

Due to the growing need to increase research focused on the Mexican economy, based on machine learning, it is important to include, in the solutions, algorithms and varied approaches for understanding trade retail. The purpose is to identify evidence in the form of patterns from the data, with which various informed and thoughtful analyzes can be carried out on the current situation of retail trade and its impact on the national economy.

3 Method

The method defined for the analysis of the behavior of retail trade in Mexico was divided into four stages: a) data acquisition, b) exploratory data analysis, c) selection of variables, and d) algorithmic application.

3.1 Data Acquisition

The analyzed data were obtained from the Annual Trade Survey. This survey is based upon the economic units from the National Statistical Framework of Economic Units (MENUE, by its acronym in Spanish), supplied by the Mexican Business Statistical Registry (RENEM, by its acronym in Spanish) with referenced design variables [28].

The data source corresponds to open data available through the official website of the INEGI[1] , which is made up of data matrices on prime economic indicators of commercial activity by sector, subsector, and branch of economic activity at the national level. The global information is made up of 40 branches of economic activity and is made up of 2134549 businesses in the commercial sector. Of these, 18 branches belong to the wholesale trade (comprising 126933 business) and 22 to the retail trade (comprising 2007616 business), being, the latter, and the object of study in this research work.

Furthermore, the North American Industrial determines the variables used Classification System, which allows us to create clusters systematically, always under

[1] www.inegi.org.mx/app/descarga/ficha.html?tit=110334&ag=0&f=csv

Table 1. Aggregation levels of economic activities by sector, subsector, and industrial branch.

Code	Levels of aggregation
4611	Retail trade of groceries and food products
4612	Retail trade of beverages, ice, and tobacco
4621	Retail trade in self-service stores
4622	Retail trade in department stores
4631	Retail trade of textile products, except apparel
4632	Retail trade of clothing, costume jewelry, and clothing accessories
4633	Retail trade of footwear
4641	Retail trade of health care items
4651	Retail trade of perfumery and jewelry
4652	Retail trade of entertainment articles
4653	Retail trade of stationery, books, magazines, and newspapers
4659	Retail trade of pets, gifts, religious articles, disposables, handicrafts, and other articles for personal use
4661	Retail trade of household furniture and other household goods
4662	Retail trade of furniture, computer equipment and accessories, telephones, and others communication devices
4663	Retail sale of articles for interior decoration
4664	Retail trade of used goods
4671	Retail trade of hardware, plumbing, and glassware
4681	Retail trade of cars and trucks
4682	Retail trade of parts and spare parts for automobiles, vans, and trucks
4683	Retail trade of motorcycles and other motor vehicles
4684	Retail trade of fuels, oils, and lubricating grease
4691	Retail trade exclusively through the Internet, and printed catalogs, television, and similar

the same logic, which helps to avoid controversies and errors of interpretation [29]. Further, within the retail trade, the stratification by a number of workers in each company according to INEGI is defined as [30]: i) micro (up to 10 people), ii) small (11 to 30 people), and iii) medium (31 to 100 people).

3.2 Exploratory Data Analysis

The analysis period was from 2016 to 2019, since in 2020 the final figures were published as part of the 2019 Economic Censuses. In this sense, an exploratory analysis was initially carried out on the data set, which was useful to know the data and understand its main characteristics. Thus, based on data exploration, it was observed that the data structure is made up of 61 variables that represent economic activities.

The data recorded are non-negative numbers, which were grouped into number of establishments, number of people, and thousands of pesos (national currency). Also, there are no null values. It was also observed that there are no out-of-range values. Table 1 shows the levels of aggregation, where the first two digits correspond to the sector (46), the first three digits to the subsector, and the four digits as a whole to the branch of economic activity.

The degree of linear relationship between pairs of variables was also measured. This information was structured in a correlation matrix, finding mostly a weak relationship

between the variables. However, those variables that presented a certain relationship were those belonging to the same economic activity. In addition, it was necessary to carry out a selection of variables. This selection allowed focusing the analysis on the significant variables of different economic activities of the retail trade.

3.3 Feature Selection

From each of the 22 retail trade branches, shown in Table 1, a set of 58 variables was obtained, from which categorical variables were filtered, and the year was discarded because it inherently already represents a clustering of data, which is to be avoided, since the objective is to obtain a segmentation that combines all the variables through the measurement of similarities between the available elements. In this sense, the selected variables offer information on the following categories:

1. Stratum of the business: micro, small or medium.
2. Types of establishments: auxiliary or commercial (number of establishments).
3. Dependent and non-dependent personnel: women and men (number of people).

The other selected variables represent different significant activities of the retail trade, such as: a) consumption of goods and services; b) taxes on the activity; c) net sales; and d) fixed assets. These financial activities represent the set of operations that are executed in the supply and demand market, whose path is the acquisition of income and the realization of expenses.

3.4 Algorithm Application

For the cluster analysis, the K-means algorithm was used due to its functionality and efficiency characteristics, where through a method, such as the Elbow, it is possible to define the appropriate number of clusters into which the data vectors should be divided (elements), which make up the data set.

Furthermore, through this algorithm optimization problem are dealt with, in which the elements are distributed in K clusters so that the sum of the internal variances of all of them is as low as possible. Thus, in order to find similarities in the different branches of retail trade, the algorithm was implemented in Python, in such a way that clusters were generated based on the following process of assigning elements and updating centroids:

Start: centroids chosen during each iteration were established randomly for the formation of clusters.

1. Assignment: each data point (vector) was assigned to its nearest centroid.
2. Update: the average of all assigned points in the cluster was calculated to set the new centroid.
3. Repeat: steps 2 and 3 were repeated iteratively until the centroids no longer changed.

The elements were assigned employing the minimum distances, measured through the Euclidean distance, between each element and the centroids, achieving thus, a high intra-cluster similarity and low inter-cluster similarity. The equation of the Euclidean distance is as follows:

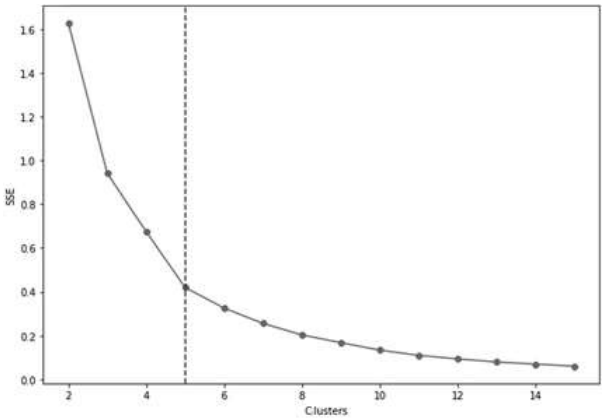


Fig. 1. Elbow method for identifying the adequate number of clusters.

$$\mathrm{dist}(p, c) = \sqrt{\sum_{i=1}^{n} (p_i - c_i)^2}\,, \tag{1}$$

where, $p = \{p_1, \ldots, p_n\}$ are the elements of the data set and $c = \{c_1, \ldots, c_k\}$ corresponds to the centroids. Therefore, by virtue of the need for a priori knowledge about the adequate number of clusters, a range of k configurations was established for the implementation of the algorithm.

This range allowed the algorithm to be run iteratively to obtain the clusters. Subsequently, based on the resulting categorization, the sum of the squared error (SSE) between each element of the formed cluster and its closest centroid was calculated. This SSE estimate was for each configuration of k based on the following equation:

$$\mathrm{SSE} = \sum_{k=1}^{k} \mathrm{dist}(p_i, c_i) = \sum_{k=1}^{k} \sum_{p_i \in C_k} (p_i - c_i)^2\,. \tag{2}$$

Since this is a measure of error, the goal of K-means is to try to minimize this value. This measurement of error is used to carry out the elbow method, in which a curve is drawn with the values obtained from SSE to find an inflection point (elbow), through which the optimal number of clusters to be analyzed is established. Figure 1 shows the layout of this curve, in which it is observed that the elbow effect suddenly changes its orientation in k equal to 5.

4 Results

From the results obtained, a differentiated segmentation was observed for the evaluation period (2016-2019). Thus, based on the internal similarities of the clusters and the values of their centroids, Table 2 shows a summary of their most significant

characteristics, determined by the aspects in which each cluster stands out for having greater or lesser participation in the retail trade activities; stratum (micro, small and medium-sized business), types of establishments and personnel (dependent and non-dependent); and consumption, taxes, sales and fixed assets.

Cluster 1 is distinguished by having the least amount of money directed to the tax payment levied on commercial activity, and specific taxes on the products sold. Also characterized by the lack of control and tax burden. A case is the businesses that sell through the Internet, which, despite the increase in the consumption of products and services through digital platforms, the payment of their taxes is still low. This causes an impact on tax collection, generating administrative and control weaknesses. According to the economic study carried out by the OECD, tax revenues in this branch continue to be low and fiscal policy has a lower redistributive impact.

This causes an impact on tax collection, generating administrative and control weaknesses for the payment of taxes. Contrary to the previous cluster, Cluster 2 is characterized by allocating the largest amount of resources to consumption, tax payments, sales, and fixed assets. This behavior is because the retail trade, for example, in self-service stores, it stands out for having a higher productivity and more efficient distribution of products. Cluster 3 stands out for being made up of the grocery and food retail trade, as one of the most common industrial branches in Mexican society, where competition is local.

In addition, this type of commerce consists mainly of micro-businesses, since they have fewer barriers thanks to the behavior of consumers, who go to the stores closest to their home to obtain better prices, greater variety, or another benefit. Cluster 4 has differences between the industrial branches that comprise it, since they do not follow the same trend throughout the period analyzed.

This may be due to the fact that these branches have undergone a process of transformation in recent years, for example, the retail trade of hardware stores and glass items has presented significant declines with respect to foreign investment. For its part, Cluster 5 is characterized by having increased participation of small and medium-sized businesses.

This is important because, during 2020, the same trend continued according to the information provided by Data Mexico, there was an in-crease of 15.1% in small businesses and 10.6% in medium businesses, this compared to the previous year (2019). Meanwhile, the micro businesses continued with a fall of 2.45%.

Regarding the participation in the different financial items and activities, it was observed that Cluster 1, unlike Cluster 2, is the one with the most industrial branches and that they have lower participation in all financial activities. This confirms the characteristics mentioned in the background section regarding SMEs and how they survive in the midst of an aggressive and demanding market.

It was also observed that clusters 3 and 5 have greater participation in the three strata (micro, small and medium business), compared to clusters 1, 2, and 4. On the other side, in the conformation of the five clusters, there are industrial branches that have participated in more than one cluster, such as:

1. Retail trade of the furniture, computer equipment, and accessories, telephones, and other communication devices,
2. Retail trade of parts and spare parts for automobiles, vans, and trucks; and

Table 2. Summary of the clusters obtained.

Cluster	Characteristics
1	**Industrial branch** – Beverages, ice, and tobacco | Textile products, except apparel | Clothing, costume jewelry and clothing accessories | Footwear | Perfumery and jewelry articles | Entertainment articles | Stationery, books, magazines, and newspapers | Pets, gifts, religious articles, disposables, handicrafts and other articles for personal use | Household furniture and other household goods | Furniture, computer equipment and accessories, telephones, and other communication devices (2016 and 2017) | Interior decorating articles | Used goods | Parts and spare parts for automobiles, vans, and trucks (2016 and 2017) | Motorcycles and other motor vehicles | Retail trade exclusively through the Internet, and printed catalogs, television and similar. **Stratum, types of establishments and personnel (dependent and non-dependent)** – The lowest number of small and medium-sized companies. – The lowest number of auxiliary establishments. – The lowest number of dependent personnel (men) and non-dependent personnel (men and women). **Consumption, taxes, sales, and fixed assets** – The lowest consumption of merchandise, materials, raw and auxiliary materials. – The lowest number of taxes levied on the activity and specific to the products. – The lowest net sales of merchandise, manufactured products, services rendered, rental of movable and immovable property. – The lowest purchase and sale of machinery and production equipment, real estate, transportation units and equipment, computer and peripheral equipment, furniture, office equipment, and other fixed assets.
2	**Industrial branch** – Retail trade in department stores. **Stratum, types of establishments and personnel (dependent and non-dependent)** – The greatest participation of non-dependent personnel (men and women). **Consumption, taxes, sales, and fixed assets** – The biggest amount of money for consumption of merchandise, fuels and lubricants, electrical energy, containers, and packaging; payments for the rental of movable and immovable property, personnel non-dependent, advertising, and communication services. – The highest number of specific taxes on products. – The highest net sale of merchandise, manufactured products, income from services rendered, rental of movable and immovable property. – The greatest purchase and sale of machinery and production equipment, real estate, computer and peripheral equipment, furniture, office equipment, and other fixed assets.
3	**Industrial branch** – Groceries and food products | Hardware, plumbing, and glassware products (2018 and 2019). **Stratum, types of establishments and personnel (dependent and non-dependent)** – The greatest number of micro-companies. – The greatest number of auxiliary and commercial establishments. – The greatest number of dependent personnel (men and women). **Consumption, taxes, sales, and fixed assets** – The greatest number of materials consumed for services rendered, raw and auxiliary materials. – The greatest number of taxes levied on the activity. – The lowest consignment and commission income.
4	**Industrial branch** – Retail trade in department stores | Health care items | Hardware, plumbing and glassware products (2016 and 2017) | Cars and trucks | Furniture, computer equipment and accessories, telephones and other communication devices (2018 and 2019) | Parts and spare parts for automobiles, vans, and trucks (2018 and 2019). **Stratum, types of establishments and personnel (dependent and non-dependent)** – The lowest number of small companies. **Consumption, taxes, sales, and fixed assets** – The greatest consignment and commission income.
5	**Industrial branch** – Retail trade of fuels, oils and lubricating grease. **Stratum, types of establishments and personnel (dependent and non-dependent)** – The greatest participation of small and medium-sized companies. – The lowest number of commercial establishments. – The lowest number of dependent personnel (women). **Consumption, taxes, sales, and fixed assets** – The greatest purchase and sale of transportation units and equipment.

3. Retail trade of hardware and glass items. This behavior is due to the years of analysis, from 2016 to 2019, associated with the 22 branches of the retail trade.

Consequently, it is possible to observe that the different trends that prevailed over the years, in the different industrial branches, are a reflection of the economic activity of the country and its industrial sectors. Activities such as imports directly affect merchants' suppliers. On the other hand, foreign investment in different industrial sectors also has a direct impact on retail businesses.

Sectors such as manufacturing industries, food and beverage preparation, wholesale trade, real estate services, construction, mining, agriculture, animal husbandry and exploitation, forestry, fishing, and hunting; as well as generation, transmission, distribution, and commercialization of electrical energy, supply of water and natural gas through pipelines to the final consumer (in this case for use in economic units dedicated to retail trade), have a great impact on the items analyzed for each industrial branch.

Certainly, it is important to analyze the market in which the retail trade operates from the point of view of the consumer and how its decisions affect the retailer. The results obtained provide the basis for understanding the power that the consumer has within commerce in Mexico. For example, today supermarkets have displaced retailers, causing consumers to prefer supermarkets instead of small businesses.

This is due to the ability of large companies to offer lower prices and offer a huge variety of products. These characteristics have led some retailers out of business. To avoid business closures, the participation of retail businesses in tax regimes should be promoted.

For example, the Ministry of Finance and Public Credit, together with the Tax Administration Service, must continue with the incorporation of SMEs to the Tax Incorporation Regime (RIF), to obtain benefits, such as discounts on income tax (ISR), deduction of payments, issuance of electronic invoicing, social security, financing or credits. However, pro-retail policies for many entrepreneurs are partial solutions, causing them to fail or survive in the midst of an aggressive and demanding market.

5 Conclusions

The retail trade has experienced risks and difficulties over the years that impede its development and growth. Since this type of trade is an important part of the economic and social stability of a country, it is essential to understand its behavior, which, hand in hand with specialized machine learning algorithms, is possible to do so. The importance of data in economics is useful for the development of social impact studies that provide relevant information. Under this idea, working with open data from the retail trade represented a significant challenge, since key variables were identified for obtaining useful results on the population dedicated to the retail trade.

In line with the above, it is essential to highlight the importance of retail trade within the Mexican economy, considering that it is the sector to which many SMEs are dedicated and, in terms of GDP, it is the source of numerous jobs. However, it is also struggling to survive in the market. These SMEs subsist in the midst of the vulnerability of their businesses since they lack financial management, which does not allow them to develop within a demanding market. They also do not have the resources to update

their technology. From the foregoing, the need for policies to benefit workers dedicated to the retail trade without requiring complex administrative procedures is derived.

The incorporation of retail businesses into tax regimes is a solution that seeks the protection and well-being of workers in any eventuality. However, it is a complex activity to manage for most workers, so focusing efforts to promote support programs for micro, small and medium-sized businesses is essential for their growth and economic development. On the other side, in order to identify strengths, weaknesses, opportunities, threats and even risks that the retail trade faces, the use of machine learning algorithms becomes essential, since they provide a way to the resolution of specific problems. For this reason, the application of the K-means clustering algorithm allowed a better understanding of the behavior of the retail trade.

Undoubtedly, making use of clustering algorithms proved to be a powerful tool for observing the dynamism over the years of the trade sector, thus confirming the importance of applying machine learning as a support tool in data analytics in the field of economics and, with this, to know the growth economic of Mexico. It was observed that the retail trade has experienced risks and difficulties that impede its development and growth, therefore, measures must be established to guarantee support plans, regardless of their size; provide guidance on the shortage of skilled labor; advise on subsistence in the sector after the crisis due to the COVID-19 pandemic; and diversify sales channels, especially by helping small physical retailers sell online.

As future work, it is intended to make a new analysis with updated information to enrich the results obtained. This may be important due to the behavior of the COVID-19 pandemic and its impact on the retail trade, which, according to sources consulted, registered a significant drop, compared to months prior to said pandemic. In addition, it would be relevant to focus efforts on the visualization of results through visual resources such as graphics, maps or a graphical interface aimed at interested users and thus contribute to decision-making, or information analysis in this area.

References

1. Molero-Castillo G., Maldonado-Hernández G., Mezura-Godoy C., Benítez-Guerrero E.: Interactive system for the analysis of academic achievement at the upper-middle education in Mexico. Computación y Sistemas, vol. 22, no. 1, pp. 223–233 (2018) doi: 10.13053/CyS-22-1-2773
2. Montuschi, L.: Datos, información y conocimiento. De la sociedad de la información a la sociedad del conocimiento. Universidad del CEMA, vol. 192, no. 6, pp. 2–32 (2001)
3. Piedras, E.: Industrias y patrimonio cultural en el desarrollo económico de México. Cuicuilco, vol. 13, no. 38, pp. 29–46 (2006)
4. Lee, I., Shin, Y. J.: Machine learning for enterprises: Applications, algorithm selection, and challenges. Business Horizons, vol. 63, no. 2, pp. 157–170 (2020)
5. Hansen, S.: Aplicación del aprendizaje automático al análisis económico y la formulación de políticas. Papeles de economía española, vol. 157, pp. 216–234 (2018)
6. Kumar, M. R., Venkatesh, J., Rahman, A. M.: Data mining and machine learning in retail business: Developing efficiencies for better customer retention. Journal of Ambient Intelligence and Humanized Computing, pp. 1–13 (2021) doi: 10.1007/ s12652-020-02711-7
7. Quiroga-Persivale, G.: ¿Qué es la inteligencia artificial y cómo se aplica en los negocios? (2018)

8. Mathur, P.: Overview of machine learning in retail. Machine Learning Applications Using Python. Apress, Berkeley, CA, pp. 147–157 (2019) doi: 10.10 07/978-1-4842-378787
9. Comercio al por menor. http://centro.paot.org.mx/documentos/inegi/comercio menor.pdf
10. INEGI: Producto interno bruto trimestral: Por actividad económica.
11. INEGI: Glosario.
12. Arana, D.: Pymes mexicanas, un panorama para 2018 (2017)
13. Ávila-Lugo, J.: Introducción a la economía. Ed. Plaza y Valdez, México, pp. 390, ISBN: 970-722-256-5 (2007)
14. INEGI: Clasificación para actividades económicas. Encuesta Nacional de Ocupación y Empleo (ENOE), www.inegi.org.mx
15. Gabel, S., Guhl, D., Klapper, D.: P2V-MAP: Mapping market structures for large retail assortments. Journal of Marketing Research, vol. 56, no. 4, pp. 557–580 (2019) doi: /10.1177/00222437198336
16. INADEM: Conflictos en el emprendimiento. www.inadem.gob.mx/conflictos-en-el-emprendimiento
17. ENOE: Resultados del tercer trimestre de 2020. www.inegi.org.mx/contenidos/ programas/enoe/15ymas/doc/enoe_n_presentacion_ejecutiva_trim3.pdf
18. INEGI: Encuesta nacional de ocupación y empleo (ENOE), población de 15 años y más de edad. www.inegi.org.mx/programas/enoe/15ymas/
19. INEGI: Encuesta anual del comercio 2019. www.inegi.org.mx/prog ramas/eac/2013/
20. OMC: México y la OMC. Available in: www.wto.org/spanish/thewto_s/countries s/mexico_s.htm
21. Palma Méndez, J. T., Marín Morales, R.: Inteligencia artificial: Métodos, técnicas y aplicaciones. Madrid, MacGraw-Hill, pp. 1022 (2008)
22. Rouhiainen, L.: Inteligencia Artificial. Madrid, Alienta Editorial, https://static0 planetadelibroscom.cdnstatics.com/libros_contenido_extra/40/39308_Inteligencia_artificia l.pdf
23. Cambronero C. G., Moreno, I. G.: Algoritmos de aprendizaje: KNN & Kmeans, www.it.uc3m.es/~jvillena/irc/practicas/08-09/06.pdf
24. Pla, D., Pascual, F., Sánchez, S.: Algoritmos de agrupamiento. Métodos Informáticos Avanzados, pp. 164–174 (2007)
25. Soto, A. J., Ponzoni, I., Vazquez, G. E.: Análisis numérico de diferentes criterios de similitud en algoritmos de clustering. Mecánica Computacional, pp. 993–1012 (2006)
26. Pérez, M.: Aplicación de Kmeans y SOM.
27. Prasanna, R., Jayasundara, J., Naradda Gamage, S., Ekanayake, E., Rajapakshe, P., Abeyrathne, G.: Sustainability of SMEs in the Competition: A systemic review on technological challenges and SME performance. Journal of Open Innovation: Technology, Market, and Complexity, vol. 5, no. 4, pp. 1–18 (2019)
28. EAC: Síntesis metodológica: Encuestas Económicas Nacionales
29. SCIAN: NAICS–SCIAN. https://naics-scian.inegi.org.mx/naics_scian/defaul t_e.aspx
30. INEGI: Censos económicos 2019. Micro, pequeña, mediana y gran empresa, www.inegi.org.mx/contenidos/productos/prod_serv/contenidos/espanol/bvinegi/productos/ nueva_estruc/702825198657.pdf

ISSN 1870-4069

Feet Point Cloud Orientation, Localization and Semantic Segmentation

Ricardo C. Villarreal-Calva, Ponciano J. Escamilla-Ambrosio, Juan Humberto Sossa Azuela

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{rvillarrealc2020, pescamilla, hsossa}@cic.ipn.mx

Abstract. In this work, an algorithm towards automatic feet measurements extraction for personalized footwear design is presented. The algorithm performs orientation, localization and semantic segmentation of a point cloud produced by a 3D feet scan system. The algorithm input instance is a point cloud from a 3D scan of a user standing on their feet on a flat surface. The outputs are two point clouds; each one is a segmentation of the user´s left and right foot with its own coordinate system. The algorithm implementation is done with the point cloud library Open3D. It combines linear programming and techniques from Machine Learning such as RANSAC and DBSCAN together with the point cloud processing algorithm Screened Surface Poisson Reconstruction to process the localization and segmentation of the foot. The proposed method contributes to the extraction of anthropometric measures of the feet in order to be able to build personalized footwear, which is ongoing research.

Keywords: Point cloud, semantic segmentation, localization, footwear customization.

1 Introduction

Technologies available for specialized users, with time will reach the maturity required for mass consumption. Such an actual case is the light detection and ranging (LiDAR) sensors which was popularized after the DARPA urban challenge competition on self-driving cars [1]. Today, this technology is embedded into higher end tablets and smartphones produced by the company Apple®.

The LiDAR and cameras on these devices produce an RGB-D image, which is a regular image with a distance value for each pixel. Several RGB-D images can be processed to create a 3D scan of a scene in a process known as scene registration [2]. It is expected that this technology will improve the experience of online stores as users will be able to scan its whole body or parts of it and get the proper size for every product they buy online [3].

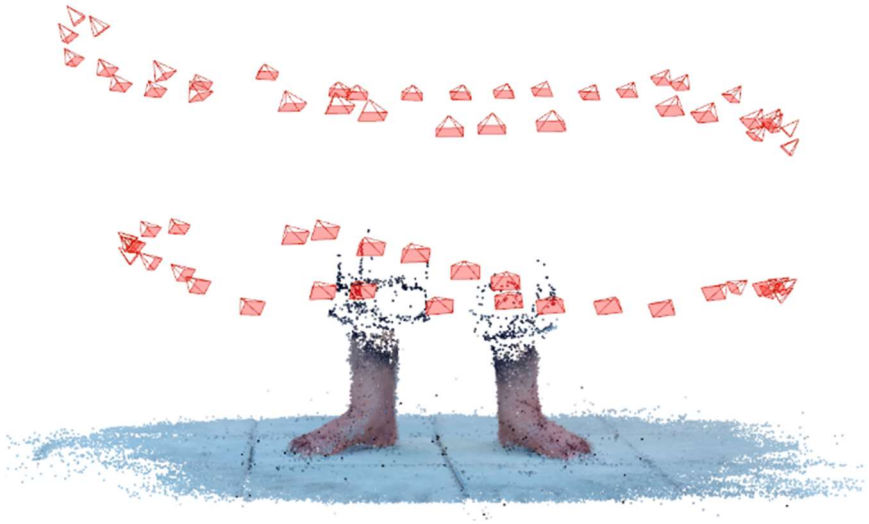


Fig. 1. Typical input point cloud from the feet scan process. For the Multi View shape representation around 60 photographs of the feet are taken; then the photographs are processed to change from Multi View to point cloud shape representation. The icons around the feet represent the position and orientation of the camera for each photograph taken.

Unfortunately, current footwear mass production provide a proper fit to 90% of customers with 3 shoe widths per length [4], so there exists an important percentage of the population that uses wrong sized shoes, which can produce foot deformities and decrease life quality over time. The present work proposes a method to do semantic segmentation of the feet scan of a person in standing position.

The method produces a point cloud and a coordinate system for each foot, so it will be possible to further process it to extract anthropometric measures such as the ones described in [5] to design personalized footwear. The outline of this work is as follow: in section two, the input instance expected for the algorithm is described. Section three presents the floor segmentation and initial alignment of the Z axis. Section four describes the foot clustering and removal of foot noise. Finally, in section five the creation of the new coordinate system for each foot is introduced.

2 Data

The algorithm requires as input a point cloud X, such that each point $x \in X$ belongs to the Euclidean space R^3. Even tough the point cloud might have color information for each point; this data is deliberately ignored, as it would require us to consider aspects like different skin tones and floor color for the segmentation.

The point clouds used for this study are obtained from Multi View 3D object representation [6], which consists of around 60 photographs of the subject to scan. Each photograph is processed through the photogrammetry software COLMAP [7, 8] to transform them from Multi View to Point Cloud shape representation.

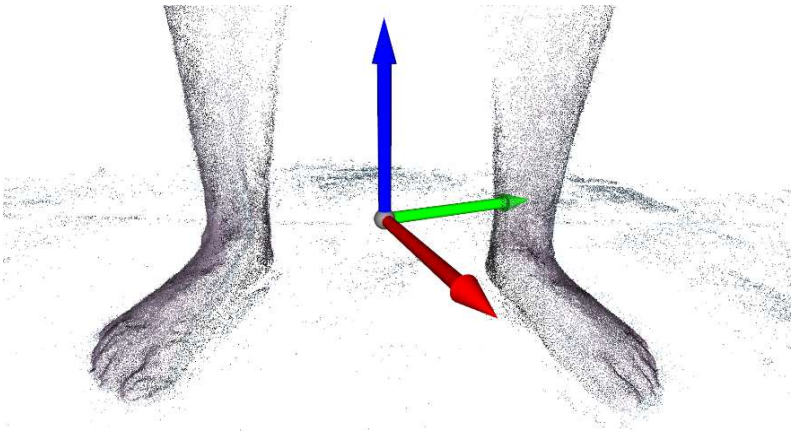


Fig. 2. Point cloud after removal of all points identified to belong to the floor's plane and alignment of the floor to the Z-axis with its direction pointing upward from the floor. The red arrow represents the X-axis, the green arrow the Y-axis and the blue arrow the Z-axis.

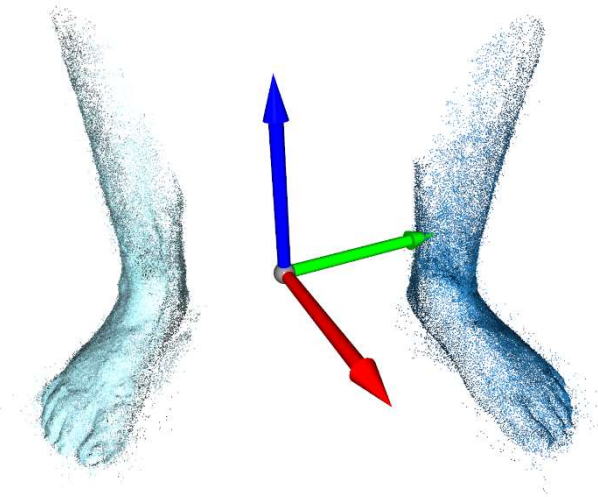


Fig. 3. Points of the two largest clusters found by DBSCAN (without points from the noise cluster).

This method assigns an arbitrary coordinate system to the point cloud, which is dimensionless; the cardinality of the point cloud is in the order of hundreds of thousands of points. The 3D feet scan must comply with the following characteristics:

- The person at time of the 3D scan must be standing on a flat surface (knee above heel).
- The scan must cover at least 1/2 of the knee height and up to half meter.
- The scan must have at least one square meter of floor.
- The scan must not have points below the floor.

– Feet must be free of strong amputations.
– Foot must have a separation of at least 15 cm between them.

3 Floor Segmentation and Z Axis Alignment

The first step towards foot localization is to identify the points that belong to the floor. Those points lie over a plane, which needs to be characterized to align it to the Z-axis. We also need to ensure that the positive Z-axis direction point upward from the floor.

3.1 Floor Segmentation

From Figure 1 we can see that we could do the floor segmentation based on color data. We could choose blue points and classify them as floor, but such strategy would fail with other floor colors. We could also do the segmentation based on skin color, but this would have the same weakness as people around the world have different skin tones.

As we want to create a robust algorithm for the segmentation, the strategy is to identify every point that belongs to the floor through the use of the Random Sample Consensus (RANSAC) [9] plane segmentation algorithm implemented on the Open3D library [10].

RANSAC takes three aleatory points from the point cloud to define a plane, then counts the inlier points (those closer than a threshold to the plane) and repeats this process several times. The set of points from the plane with more inlier points is used to find the plane such that the summed squared distances from the plane to all points is minimized.

The fitted plane is returned as (a, b, c, d) such that for each inlier point (x, y, z) we have $(ax + by + cz + d \approx 0)$. The algorithm also returns a list of indices of the inlier points. RANSAC algorithm requires three arguments: the maximum distance (δ) a point can have to an estimated plane to be considered an inlier, the number of points (n_p) that are randomly sampled to estimate a plane, and the number of times (n_t) a random plane is sampled and verified.

A worst case scenario is used to quantify the probability that RANSAC will find the floor plane; we define a box with the biggest foot dimensions found in the Mondopoint scale [11] and the maximum height of the scan. The box surface but the face touching the floor represents the area of the foot. To compute the area of the feet A_{feet} we have:

$$A_{\text{feet}} = 2(2lh + 2wh + lw), \tag{1}$$

where $l = 32\ cm$, $w = 12.2\ cm$ and $h = 50\ cm$, so $A_{\text{feet}} = 9{,}620\ cm^2$, we have the minimum floor area which is $A_{\text{floor}} = 10{,}000\ cm^2$. We define a Binomial Random Variable to calculate the probability P_f that RANSAC will choose three points from the floor to define a plane:

$$P_f = \binom{n}{k} p^k (1-p)^{n-k}, \tag{2}$$

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}, \tag{3}$$

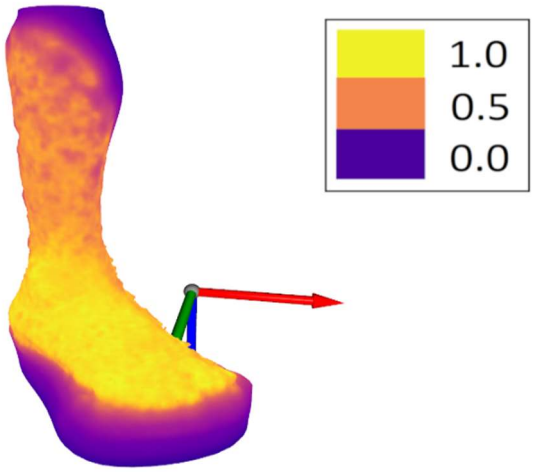


Fig. 4. Mesh for one foot displayed with a shader where violet indicates low point density and yellow high point density.

where p is the probability of choosing a point from the floor and is calculated by:

$$p = \frac{A_{\text{floor}}}{A_{\text{floor}} + A_{\text{feet}}}. \tag{4}$$

And we want to calculate the probability of three successes in three trials ($n = 3,\ k = 3$). Performing computation with the previous parameters give $p = 0.51$ and $P_f = 0.132$. As RANSAC repeats this process, by The Central Limit Theorem we can approximate the distribution of planes found from floor points with a normal density function with parameters $u = np$ and $\sigma = \sqrt{npq}$ where $n = n_t,\ p = P_f$ and $q = 1 - P_f$.

For this work $n_t = 1{,}000$, therefore, we have that RANSAC will find a mean $u = 132$ of different planes from points of the floor with a standard deviation of $\sigma = 10.7$. Therefore, RANSAC with a probability higher than 99.7% will chose the best floor plane from more than 100 plane samples from the floor.

3.2 Z Axis Alignment

The equation of the floor plane is used to rotate the point cloud such that its normal vector is parallel to the Z-axis; then a translation transformation is applied to the point cloud such that the center of the floor´s inlier points is translated to the origin of the coordinate system.

The center of the point cloud must be between the floor and the knee of the user, so if the Z coordinate of the point cloud center has a negative value, then the point cloud is rotated 180° around the X-axis; otherwise, it is left in its current configuration.

Finally, all floor's inlier points are removed from the point cloud. Figure 2 shows the point cloud once this process ends with the coordinate system´s Z-axis pointing upward from the floor and the XY plane aligned to the floor plane.

4 Foot Clustering

From Figure 2 we can see that not all floor points are removed, therefore, further processing is required. The algorithm chosen is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) implemented in Open3D. DBSCAN is a recursive algorithm that starts with an arbitrary point and counts the number of points at a distance lower than a threshold (eps), if the quantity is more than the minimum of points required to form a cluster, then a cluster is created; then close clusters are merged.

The algorithm returns for each point either its cluster label or that it is classified as noise, in Figure 3 we can observe the two largest clusters found. DBSCAN requires two parameters: eps, which defines the maximum distance between neighbors in a cluster and min points, which defines the minimum number of inlier points required to form a cluster.

4.1 Foot Noise Removal

To filter the noise around both foots, the Screened Poisson Surface Reconstruction (SPSR) is used. SPSR produces watertight surfaces from oriented point sets by (1) transforming the oriented point samples into a continuous vector field in 3D, (2) finding a scalar function whose gradients best match the vector field, and (3) extracting the appropriate isosurface [12]. The surface has different densities as the algorithm will also create triangles in areas of low point density, and even extrapolates into some areas without points.

The input point cloud to SPSR must have normal vectors, Figure 4 shows the resulting mesh with a shader where normalized point densities of the mesh are displayed with pseudo color, where yellow indicates high point density, orange midpoint density, and violet low point density.

The area under the foot clearly does not have supporting points. SPSR requires parameter depth, with high values the final mesh will have more details, lower values produce smoother surfaces. In Figure 5 we can observe the mesh without low density triangles inside an Axis Aligned Bounding Box (AABB) which serves as guide to find the proper coordinate system in further steps.

5 Foot Alignment to the Coordinate System

From Figure 5 we can observe that every AABB edges are parallel to one of the axes of the coordinate system. To find a coordinate system for each foot, some properties of the AABB will be used:

1 An AABB encloses all vertices in the input mesh with the lowest possible volume.

2 Each AABB face normal vector is parallel to some axis of the coordinate system.

First, as explained in Section 3, we know that the Z axis is parallel to the floor's plane normal vector, and its direction is pointing upward from the floor. Now, we will find the Z-axis rotation for each foot that minimizes the length of the AABB along the X-axis using linear programming; the rotation search space is $[0, \pi]$. The minimization

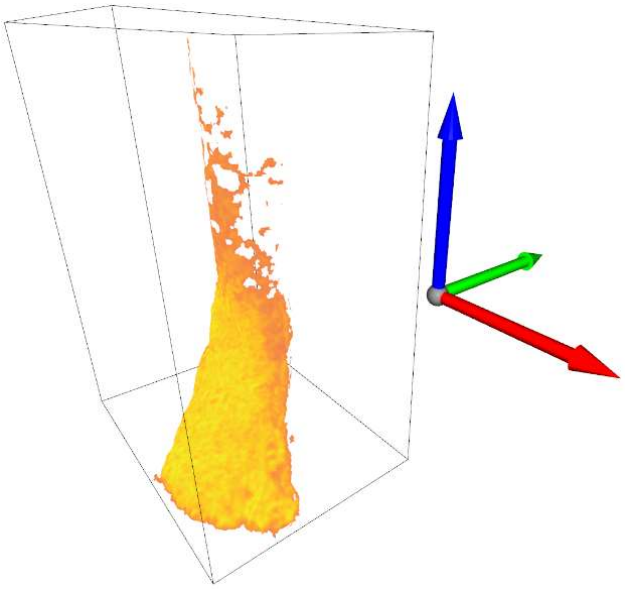


Fig. 5. Resulting mesh after removal of low-density triangles inside the AABB.

algorithm used is a modified version of [13] implemented in SciPy library [14]. The returned angles found are used to rotate each foot model around the Z-axis.

5.1 Select of the AABB Face to be Aligned to the XZ Plane

We want to position the foot such that the direction from heel to toes points towards the positive Y direction. Linear programming does not guarantee this, it just aligns the foot in the X and the Y-axes, but the solution found might be in the wrong direction, as seen in the foot on the right of Figure 6. The next algorithm is used to identify if the foot needs to be rotated to properly be aligned to the Y-axis.

1 The input instance is the point cloud cluster of one foot rotated as the linear programming output.

2 Select the two vertices of the AABB of the cluster's high-density foot mesh with lower X value and higher Z value.

3 Count the points inside a radius of length r from each vertex selected, where r is the width of the AABB.

4 Choose vertex with higher points.

5 If the vertex selected is the one with higher Y coordinate, then rotate the foot by 180° around the Z-axis.

5.2 Translation of the Foot to the Origin

At this point, we have both feet properly aligned to the Z and the Y-axes, but we still need to translate them to the origin of the coordinate system.

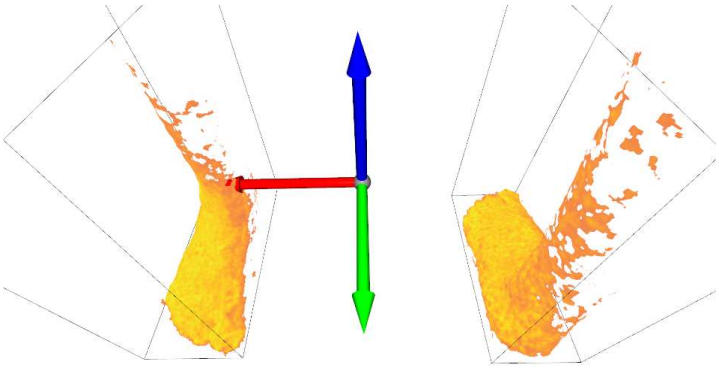


Fig. 6. Each foot mesh is rotated by angle found through linear programming. Notice that both foot lengths and widths are aligned to the Y-axis and the X-axis respectively.

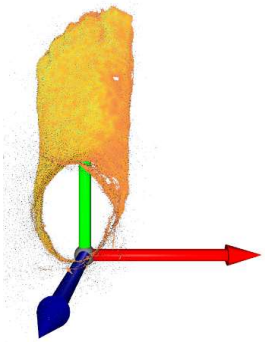


Fig. 7. Foot aligned to the coordinate system.

To do this, we use the AABB vertex with lower X, Y and Z values as anchor point of the foot and move it to the coordinate (0,0,0) and then move it half of the X width in the negative direction along the X axis. The resulting aligned foot point cloud is shown in Figure 7.

5.3 Discussion

The method presented in this work has the objective to do localization, orientation and semantic segmentation of the point cloud of the feet of a person. The evaluation at this stage is qualitative, as the photogrammetry method to recover the 3D shape produces a dimensionless point cloud; further work is required to produce a 3D shape that recovers dimension.

Future development will require creating a database of several point clouds with labels of each point to measure the precision and recall of the segmentation method. Nevertheless, the presented method is an important step towards the extraction of anthropometric measurements of a person's feet, which is a requirement to design personalized footwear.

The method to define the final precision of the anthropometric measurements is in evaluation; the two options being considered are based on Machine Learning theory and variance propagation.

The Machine Learning approach requires capturing a set of data, labeling it and defining a loss function so the error can be measured with enough confidence to state that the method generalizes properly.

The variance propagation requires to measure the error at each step of the process so an estimate of the variance of the final measurements can be calculated. As stated before, this is work in progress.

6 Conclusion

In this work, an algorithm to do point cloud segmentation and localization of foot has been presented. As first step the RANSAC algorithm to identify floor points is used and the Z axis is aligned to the plane of the floor; next with the DBSCAN algorithm most floor noise is filtered and both feet are clustered; to remove noise from the foot the SPSR algorithm is used; through linear programming the foot is aligned to the Y axis and with radius search the face of the AABB closer to the heel is detected and the correct direction in the Y axis is set. Finally, the foot shape representation is moved properly to the origin of the coordinate system.

As future work, the point cloud will be in meters so the length and width of the foot can be known, as well as other measurements required to find the proper footwear for a given user.

Acknowledgments. The authors would like to acknowledge the support provided by the Instituto Politécnico Nacional under projects 20200630 and 20210788, CONACYT under projects 65 (Fronteras de la Ciencia) and 6005 (FORDECYT-PRONACES) to carry out this research. The first author thanks CONACYT for the scholarship to undertake his master studies and to the Instituto Politécnico Nacional for the BEIFI scholarship under project 20210039.

References

1. Urmson, C., Baker, C., Dolan, J., Rybski, P., Salesky, B., Whittaker, W., Ferguson, D., Darms, M.: Autonomous Driving in Traffic: Boss and the Urban Challenge. AI Magazine, vol. 30, no. 2, pp. 17–28 (2009) doi: 10.1609/aimag.v30i2.2238
2. Dong, W., Park, J., Yang, Y., Kaess, M.: GPU Accelerated Robust Scene Reconstruction. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 7863–7870 (2019) doiI: 10.1109/IROS40897.2019.8967693
3. Sohn, J. M., Lee, S., Kim, D. E.: An exploratory study of fit and size issues with mass customized men's jackets using 3D body scan and virtual try-on technology. Textile Research Journal, vol. 90, no. 17-18 (2020) doi: 10.1177/004051 7520904927
4. Jurca, A., Žabkar, J. & Džeroski, S. Analysis of 1.2 million foot scans from North America, Europe and Asia. Scientific Reports, vol. 9, no. 10 (2019) doi: 10.1038/ s41598-019-55432-z
5. Luximon, A., Luximon, Y.: Shoe-last design innovation for better shoe fitting. Computers in Industry, vol. 60, no. 8, pp. 621–628 (2009) doi: 10.1016/j.compind. 2009.05.015

6. Gezawa. A. S, Zhang, Y., Wang, Q. C, Lei, Y. Q.: A review on deep learning approaches for 3D data representations in retrieval and classifications. IEEE Access, vol. 8, pp. 57566–57593 (2020) doi: 10.1109/access.2020.2982196
7. Schönberger, J. L., Zheng, E., Frahm, J. M., Pollefeys, M.: Pixelwise View Selection for Unstructured Multi-View Stereo. Computer Vision Lecture Notes in Computer Science, vol. 9907 (2016) doi: 10.1007/978-3-319-46487-9_31
8. Schönberger, J. L., Frahm, J. M.: Structure-from-Motion Revisited. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4104–4113 (2016)
9. Fischler, M. A, Bolles, R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, vol. 24, no. 6, pp. 381–395 (1981) doi: 10.11 45/358669.358692
10. Zhou, Q. Y, Park, J., Koltun, V.: Open3D: A Modern Library for 3D Data Processing (2018) doi: 10.48550/arXiv.1801.09847
11. Footwear - Sizing - Conversion of sizing systems. ISO/TS 19407 2015(E). Geneva, CH: International Organization for Standardization (2015)
12. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Transactions on Graphics, vol. 32, no. 3, pp. 1–13 (2013) doi: 10.1145/2487228.2 487237
13. Powell, M. J.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal, vol. 7, no. 2, pp.155–162 (1964) doi: 10.1093/comjnl/7.2.155
14. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S., Brett, M., Wilson, J., Millman, K. J., Mayorov N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J.: SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, vol. 17, no. 3, pp. 261–272 (2020) doi: 10.1038/s41592-019-0686-2

ISSN 1870-4069

3D Plant Geometry Understanding Using a CNN-Superpixel Approach

Luis A. Cundapi López[1], Carlos A. Ramírez Mendoza[1], Madain Pérez Patricio[1], German Ríos Toledo[1], J. A. de Jesús Osuna Coutiño[2]

[1] Tecnológico Nacional de México Campus Tuxtla Gutiérrez, Mexico

[2] Instituto Nacional de Astrofísica, Óptica y Electrónica, Mexico

{m13270146, m14270620, german.rt}@tuxtla.tecnm.mx, madain.pp@tuxtla.edu.mx, osuna@inaoep.mx

Abstract: Plant stress phenotyping consists of identification, classification, quantification, and prediction (ICQP) in crop stress. There are several approaches to plant stress identification. However, most of these approaches are based on the use of expert employees or invasive techniques. In general, expert employees have a good performance on different plants, but this alternative requires sufficient staff in order to guarantee quality crops. On the other hand, invasive techniques need the dismemberment of the leaves. To address this problem, an alternative is to process an image seeking to interpret areas of the images where the plant geometry may be observed, thus removing the qualified labor dependency or the crop dismemberment, but adding the challenge of having to interpret images ambiguities correctly. Motivated by the latter, we propose a new CNN-Superpixel approach for plant stress phenotyping. This strategy combines the abstraction power of deep learning and the information that provides the plant geometry. For that, our methodology has three steps. First, the plant recognition step provides the segmentation, location, and delimitation of the crop. Second, we propose a leaf detection analysis to classify and locate the boundaries between the different leaves. Finally, we use a depth sensor and the pinhole camera model to extract a 3D reconstruction.

Keywords: Plant geometry understanding, plant stress phenotyping, CNN, superpixel, deep learning.

1 Introduction

Phenotype is the observable characteristics or traits of an organism that are produced by the interaction of the genotype (the genetic constitution of an organism) and the

environment [1]. Understanding these processes that span plant's lifetime in a permanently changing environment is essential for the advancement of basic plant science [22]. Plant phenotyping is an important tool to address and understand plant environment interaction and its translation into application in crop management practices [27]. Abiotic stress includes factors such as drought, flood, salinity, radiation, high and low temperatures, among others.

Meanwhile, in biotic stress, pathogens such as bacteria, fungi, yeasts, worms (nematodes) are considered. Current approaches for accurate classification of biotic and abiotic stresses in crop research and production are predominantly visual and require specialized training. However, these techniques are subject to subjectivity and the experience of the people who perform them. In addition, the availability of visual signals allows the identification of types of stress, but these visual signals do not coincide with the symptoms determined by experts [8]. Recently, high-throughput stress phenotyping techniques have been introduced that rely primarily on remote sensing or imaging.

They are able to directly measure morphological traits, but measure physiological parameters mainly indirectly. Plant stress phenotyping is divided into four broad categories, the so-called ICQP paradigm, the acronym represents Identification, Classification, Quantification and Prediction. These four categories naturally fall into a continuum of feature extraction where increasingly more information is inferred from a given image [25].

Depending on the data acquisition devices, Plant Stress Phenotyping can be carried out in two ways: Aerial and Ground Based Sensing. The rapid development of image-based phenotyping methods based on ground-operating devices or Unmanned Aerial Vehicles (UAV) has increased our ability to evaluate traits of interestfor crop breeding in the field [10].

Studies on plant stress include those on drought stress [4], heat stress [26], salt stress [14], nutrient deficiency [18] and and biotic stress [8].

Conventional plant stress identification and classification have invariably relied on human experts identifying visual symptoms as a means of categorization [9]. According to Naik *et al*. [17], current methods for phenotypically measuring are completely visual and labor-intensive.

Naik and his collaborators reported that visual scoring is the simplest, subjective measurement that requires relatively less labor. However, it has reduced accuracy if the evaluation is made in diverse environments and by different raters [24]. Today, determination of crop stress factors using visible symptoms is still often a manual and complex task predominantly carried out by trained and experienced individuals, such as agronomists, crop scientists and plant pathologists [12].

The manual process is laborious, time-consuming and not always reproducible due to the inherently subjective nature of manual ratings, experience and interpretation [8]. It should be noted that the experience over the years is an invaluable resource. Thus, human experts can incorporate their knowledge into automated processes to improve the efficiency of these types of systems.

In the last decades, several sensors and computer vision tools have been developed and became pivotal for quantifying plant traits with increasing throughput and accuracy

[1] https://www.merriam-webster.com/dictionary/phenotype

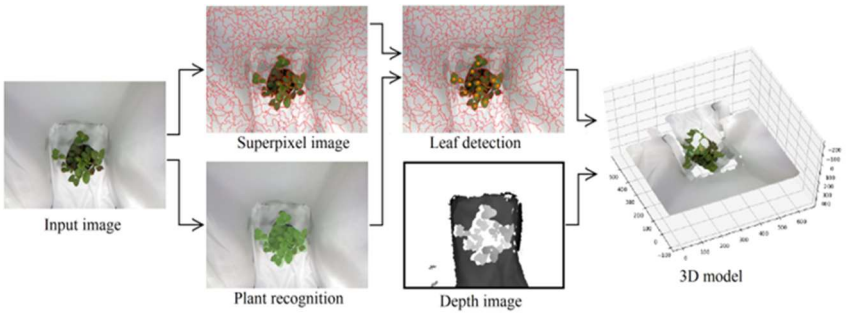


Fig. 1. Block diagram of the proposed methodology.

[22]. Computer vision is a non-contact and non-destructive sensing technology that enables multi-dimensional sensing capabilities [6].

This technology can be used to extract information from a targeted object including morphological (size, shape, texture), spectral (colour, temperature, moisture), and temporal data (growth rate, development,dynamic change of spectral and morphological states) [11]. For commercial production systems, it is more advantageous to develop a real-time plant canopy health, growth and quality monitoring system with multi-sensor platforms.

This can be achieved by a sensing system equipped with a multisensor platform moving over the canopy and ultimately using plants as 'sensors' to communicate their true status and needs [11].

Unlike conventional methods, optical imaging is advanced to measure changes caused by abiotic or biotic stressors in the plant physiology rapidly and without contact. In general, the common imaging technologies have been employed for detecting the crop stress, including digital, fluorescence, thermography, LIght Detection and Ranging(LIDAR), multispectral and hyperspectral imaging techniques [7].

Remote sensing phenotyping methods are non-destructive and non-invasive approaches [23], based mostly on the information provided by visible/near-infrared radiation reflected (or transmitted) and far-infrared(thermal) radiation emitted by the crop. Remote sensing techniques maybe deployed in situ screening for a wide range of breeding objectives,including yield potential, adaptation to abiotic (water stress, extreme temperatures, salinity) and biotic (susceptibility to pests and diseases) limiting conditions, and even quality traits [1].

Another way to measure the state of health or stress of a plant is by using laboratory techniques such as Kjeldahl method [3], a method developed for determining the nitrogen contents in organic and inorganic objects.

For example, this method was applied in [28] to measure the total nitrogen content in samples of rice plants. Kjeldahl method is the most accurate and also the most time-consuming method [19]. Also, it is invasive method since its use implies the destruction of the samples. However, it is useful in investigations where a baseline is required. Thus, it is possible to measure the efficiency of a non-invasive method such as remote-sensing technology.In recent work, there is significant progress in crop stress diagnosis

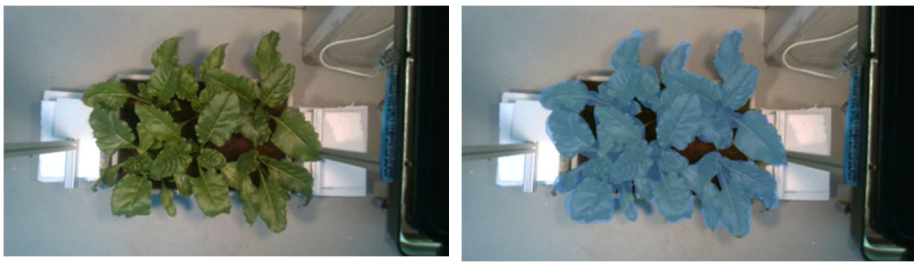


Fig. 2. (a) RGB image; (b) Semantic image.

using machine learning [2]. This was achieved via learning algorithms that learn the relationship between visual appearance and plant stress phenotyping.

Unlike the other trends (using expert employees or invasive techniques), this approach analyzes the plant without qualified labor dependency or the crop dismemberment, but adding the challenge of having to interpret images ambiguities correctly.

Water stress is one of the main causes of death in plants, it occurs in plants in response to a low water environment, where the transpiration rate exceeds the intake of water. The problem of identifying stress in plants has been extensively studied. However, the studies carried out use two-dimensional information to classify the state of a plant, posing the problem of having to extract additional information, with a range of possibilities to obtain better stress analysis results. Other works carry out stress analysis with three-dimensional models for the extraction of the plant, but use rotating tables or a set of images of different poses for its reconstruction.

To address this problem, the contribution of our work is a methodology for the extraction of a 3D model from a single image, this strategy combines the power of abstraction of deep learning and the information provided by the geometry of the plant. For this, we consider 2D and 3D information to predict the effect of water stress on growth caused by deficit of water and nutrients. Section 2 presents the proposed methodology carried out for the extraction of the 3D model and the experiments carried out, in section 3 presents the results for the evaluation of the proposed method. Finally, the results are discussed in section 4.

2 Proposed Methodology

Our methodology has three steps. First, the plant recognition step provides the segmentation, location, and delimitation of the crop. Second, we propose a leaf detection analysis to classify and locate the boundaries between the different leaves. Finally, we use a depth sensor and the pinhole camera model to extract the 3D pose. The schematic representation of the proposal is shown in Figure 1.

2.1 Input Image

We denote the RGB input image as I_p. We divide the image I_p into a grid Θ. The grid Θ consists of sections Θ_w, where w denotes the w-th section in Θ, and each Θ_w

section has a patch $\vartheta_{\varphi,\omega}$. A patch $\vartheta_{\varphi,\omega}$ is a finite set of pixels $\vartheta_{\varphi,\omega} = \{x_1, \ldots, x_u\}$, $\vartheta_{\varphi,\omega} \in \Theta$, where φ and ω are the abscissa and ordinate from the Θ grid, respectively. Pixel $\rho_{\varphi,\omega}$ is a pixel within the patch $\vartheta_{\varphi,\omega}$.

2.2 Plant Recognition

We use a CNN architecture to segment the pixels with a semantic of crop elements. This CNN learns two labels on crops (plant v^1 and no-plant v^2). Although the no-plant label does not correspond to a plant element; we use this label since we need to remove it for the next steps.

2.2.1 Training Set

In the training set of semantic segmentation, we use the "Eschikon Plant Stress Phenotyping Dataset" [13]. This dataset has spatiotemporal-spectral data pertaining to sugarbeet crop growth under no, drought, fertilizer, and weed stress conditions over two months. To obtain the training set, we divide the images of the datasets with labels (plant v^1 and no-plant v^2) in RGB images of 32×32 pixels. For example, we can obtain 630 small sections (32×32 pixels) using an image (1920×1080 pixels).

2.2.2 CNN for Semantic Segmentation

The input of the CNN is an RGB section Φ^i with a size of 32×32 pixels. In this case, we train the YOLOv4 network to learn two labels of crops (plant v^1 and no-plant v^2). Also, our program uses a sliding window with a sweep of one pixel [20]. For that, this program analyzes RGB sections of 32×32 pixels. The CNN paints the central pixel $\rho_{\varphi,\omega}$ of the analyzed RGB section (32×32 pixels). For that, the CNN paints the central pixel $\rho_{\varphi,\omega}$ of the analyzed RGB section $\vartheta_{\varphi,\omega}$ with green color if it has a plant label v^1 (See Figure 2.6). On the other hand, the CNN paints the central pixel $\rho_{\varphi,\omega}$ of the analyzed RGB section $\vartheta_{\varphi,\omega}$ with black color if it has a no-plant label v^2.

2.3 Superpixel Image

We denote the superpixel image as I_s. For the superpixel image I_s, we use the SLIC superpixel approach [19]. The purpose was to classify and locate the boundaries between the different sheets, this algorithm grouped pixels based on their similarity in color and proximity in the image plane. Where ψ_i denotes the i^{th} superpixel in an image I_s. Given an RGB image, it is first converted to the LAB color space to group pixels based on their color similarity and proximity in the image plane, improving discrimination between foreground and background sheets.

This algorithm takes as input a desired number of super pixels K of approximately the same size that is used to segment the input image with super pixels.

For the superpixel approach, we use the following parameters: desired number of superpixels = 999, number of pixel-level iterations = 9, and shape smoothing term = 5. The value of K depends on the type of images being worked on, considering the number

Table 1. Confusion matrix dataset strawberry.

		Predicted label	
		Plant	No-plant
True label	Plant	45,845	4,155
	No-plant	2,522	47,478

of objects that exist in the image, a high number of superpixels is used, in this way the algorithm allows to better locate the edges of the sheets.

The assignment and update steps are repeated until the error converges [5], but we found that 9 iterations are sufficient for the images and using K = 999, greater visualization of details is highlighted.

2.4 Leaf Detection

We use the Hough transform for the leaf detection step. We implement Hough transform to find circles in an image. For that, the Eq. 1 provides the mathematical representation of the circle. Where (a, b) is the center of the circle, and r is the radius in a fixed point (x, y). In this case, the Hough transform locates circles into the superpixel edge of our semantic segmentation. Finally, the detected circles are the leaves of the plant:

$$(x - a)^2 + (y - b)^2 = r^2. \quad (1)$$

2.5 Depth Image

Nowadays, depth-sensing technologies are widely used to scan environments or simplify challenging tasks such as object detection, pose estimation, visual tracking, among others. In this work, we use a Kinect sensor to obtain the plant depth information. For that, we denoted the depth image as D_ε. For the Kinect, we use the following parameters: image resolution= 640×480, frames per second = 12, and maximum depth = 4 meters.

2.6 3D Model Analysis

We use the basic pinhole model to extract the 3D model. This model considers the projection of a point $P(X, Y, Z)$ in space to a point $p(x, y)$ in the image plane. The relative size of an object in the image depends on the image plane distance Z and the focal length f. The focal length f is the distance between the camera center C_o (camera lens center) to the image plane. The optical center or principal point O_o is the origin of coordinates in the image plane, but in practice, it may not be. By similar

triangles, one quickly computes that the point $P(X,Y,Z)$ is mapped to the point $p(fX/Z, fY/Z, f)$ on the image plane.

We use the pinhole model with the image plane information and the depth of the sensor to compute the 3D recovery in crops. In our extraction, we convert the information in meters. For that, we divided the scale factor k with the maximum RGB value (255) and multiplied by a depth z (Subsubsection 2.5). The scale factor k is the maximum depth of the Kinect sensor (Subsubsection 2.5). For example, considering a maximum depth k of 4 meters and a depth estimation z of 128 in grayscale, Z is approximately 2 meters. The Eq. 2-4 compute the coordinates (X,Y,Z) of a point in the space:

$$Z = \frac{k \cdot z}{255}, \tag{2}$$

$$X = \frac{x \cdot Z}{f}, \tag{3}$$

$$Y = \frac{y \cdot Z}{f}. \tag{4}$$

Finally, using the 3D plant model, we calculate the 3D centroid of the detected leaves (Subsubsection 2.4). This centroid provides a 3D compact representation of the leaf. For that, we use a simplification of the intensity centroid. This simplification of intensity centroid obtains the central point by 3D leaf extracted. Defining the moments as:

$$m^j_{p,q,g} = \sum_{X,Y,Z}^{w} X^p Y^q Z^g, \tag{5}$$

where j denotes the j^{th} 3D leaf extracted, and w is the number of pixel projections by leaf. On the other hand, (p,q,g) are the orders of the moment $m^j_{p,q,g}$ (we use an order of 0 or 1). Finally, we determined the intensity centroid as:

$$C^j = \left(\frac{m^j_{1,0,0}}{m^j_{0,0,0}}, \frac{m^j_{0,1,0}}{m^j_{0,0,0}}, \frac{m^j_{0,0,1}}{m^j_{0,0,0}} \right). \tag{6}$$

2.7 Crop Irrigation and Fertigation

The experiment and acclimatization of the plants was carried out in a prototype built with a drip irrigation system with average conditions of temperature of 20 ° C and relative humidity of 60%. The prototype automatically controls 4 12v water pumps with a power of 19 watts and a current of 1A, and 2 incandescent lamps of 150 watts through a mobile application. The lamps were turned on daily from 6:00 a.m. to 6:00 p.m. and the pumps were activated at 8:00 p.m. Twelve strawberry plants organized in four groups called A, B, C and D, each with 3 plants, were placed in the prototype. Groups A and B were used to simulate stress conditions due to lack of water, while C and D were used to simulate stress conditions due to lack of nutrients. Group A was watered with a dose of 250 ml of water and group B with 150 ml of water every 2 days.

Table 1. Semantic segmentation evaluation.

N. images	Dataset sugarbeet			Dataset strawberry		
	Precision	Recall	F1-score	Precision	Recall	F1-score
15,000	0.75	0.78	0.76	0.80	0.85	0.82
20,000	0.78	0.80	0.78	0.85	0.91	0.87
25,000	0.83	0.85	0.83	0.90	0.93	0.91
30,000	0.85	0.90	0.87	0.91	0.93	0.91
40,000	0.88	0.93	0.90	0.91	0.94	0.92
50,000	0.91	0.94	0.92	0.93	0.95	0.93

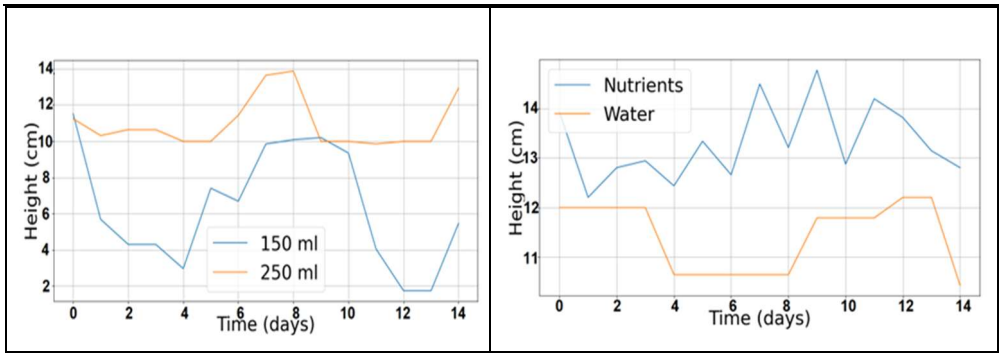


Fig. 3. (a) Water stress grid ; (b) Nutrient stress grid.

Table 2. 3D model evaluation.

RMS(x)	RMS(y)	RMS(z)	Average
0.013416	0.029814	0.007843	0.03697

The plants of group C were watered with 250 ml of water and a 17% nutrient mixture (nitrogen, phosphorus and potassium) and the plants of group D were watered only with 250 ml of water.

In the works related to water stress in plants, the observation period of the experiments is concluded when the plant shows symptoms of wilting causing discoloration or dryness in the leaves, in this way the loss of the studied crop is avoided [15]. Therefore, our observation period for both experiments was 15 days, since in later days the plants showed wilting.

To collect the images in color (RGB) and depth, a Kinect sensor placed at a distance of 90 cm above the foliage of the plants was used. The capture of images was carried out during the 15 days at 1:00 p.m. due to the lighting conditions.

A daily image was taken of a total of 12 plants, this because the plant's reaction to stress took at least one day, in this way 180 images were obtained with a resolution of 640x480 pixels.

3 Results

The problem was addressed as a binary classification problem (*plant* and *no-plant* classes), the positive class was *plant*. Binary classifier predicts the instances of the test set as positive or negative and produces four outcomes: True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP). Table 1 shows the results obtained in each prediction made by CNN training with 50,000 sections of 32x32 pixels in each class. It is observed that the network correctly predicted 45,845 sections, 92% of the total of the plant class, as parts that corresponded to the leaf and 47,478 sections, 95% of the total of the non-plant class, were correctly detected.

We use two different datasets to evaluate semantic segmentation ("Eschikon Plant Stress Phenotyping" dataset [13] and a proposed dataset). The proposed dataset was obtained from strawberry plants with 180 images with a resolution of 640x480 pixels. Table 2 shows the result of training the network with different amounts of images.

In it, the metrics are compared according to the recognition and segmentation obtained by CNN in the image sets, obtaining better results with ours. Experiments with 50,000 images showed the highest segmentation and detection precision in the leaves of the plant.

Figure 3 two different graphs are observed that show the behavior of the groups of plants from the applied irrigation, these heights were obtained from the 3D model. Figure 3 a) shows the heights of the plants watered with water in 2 different doses (250 ml and 150 ml) are shown.

The height of the leaves is measured relative to the ground. The better the plant is watered, the more the height of the leaves increases. This can be observed in the group that was watered with the 250 ml dose, the height remains constant or increases up to 2 centimeters, but its height never decreases.

Figure 3 b) shows a comparison of heights between a group of plants irrigated with nutrients and another group irrigated only with water, both groups with 250 ml.

It is observed that the height increases as a function of the soil as the days go by when a plant is watered with enough nutrients that help the plant to grow. By irrigating the plants only with water, they maintain their normal development, however the height of their leaves is maintained or even decreased.

The mean square error (RMS) determines how much the actual data differs from the predictions made by a model. Table 3 shows the RMS obtained by comparing the centroids (X, Y, Z) of the ground truth with respect to the centroids (X, Y, Z) of the CNN.

4 Conclusions

In this work, we have introduced a new CNN-Superpixel approach for 3D plant geometry understanding. Our strategy was to divide and simplify the 3D plant extraction process.

This strategy combines the abstraction power of deep learning and the information that provides crop geometry. For that, our methodology has three steps. First, the plant recognition step provides the segmentation, location, and delimitation of the crop. Second, we propose a leaf detection analysis to classify and locate the boundaries

between the different leaves. Third, we use a depth sensor and the pinhole camera model to extract a 3D reconstruction.

The quantitative experiments were conformed of the plant recognition (semantic segmentation) and its 3D extraction. In the recognition evaluation, we used two datasets that provide different crops ("Eschikon Plant Stress Phenotyping" dataset [?] and a proposed dataset). For that, we analyzed two labels of crops (plant and no-plant).

For example, our plant segmentation had an average $recall$ of 0.945, i.e., considering the ground-truth, we recognized 94.5%. On the other hand, the plant segmentation had an average $precision$ of 0.92, i.e., considering the semantic segmentation, we segmented 92.0% correctly. The segmentation is fundamental since the plant recognition is proportional to the precision of 3D extraction on the (X,Y) axis of the 3D model.

Finally, for the 3D plant extraction evaluation, we use our proposed dataset. We used the RMS error for the quantitative evaluation.

The Mean Square Error (RMS) determines how much the actual data differs from the predictions made by a model. For that, we compared the centroids (X, Y, Z) of each leaf in the 3D model of the ground truth with our centroids (X, Y, Z).

In this experiment, we have an average RMS error (Z) of 0.007843, i.e., an error of 0.007 centimeters in 4 meters. On the other hand, considering the coordinates (X, Y, Z) in the extraction, we had an average RMS (X, Y, Z) of 0.03697.

References

1. Ali, A. A.: Maize productivity in the new millennium. Hassan Awaad, MohamedAbu-hashim, Abdelazim Negm (editors), pp. 509 (2021)
2. Anami, B. S., Malvade, N. N., Palaiah, S.: Classification of yield affecting biotic andabiotic paddy crop stresses using field images. Information Processing in Agricul-ture, vol. 7, no. 2, pp. 272–285 (2020)
3. Bellomonte, G., Costantini, A., Giammarioli, S.: Comparison of modified automaticdumas method and the traditional kjeldahl method for nitrogen determination ininfant food. Journal of the Association of Official Analytical Chemists, vol 70, no. 2, pp. 227–229 (1987)
4. Clauw, P., Coppens, F., De Beuf, K., Dhondt, S., van Daele, T., Maleux, K., Storme, V., Clement, L., Gonzalez, N., Inze, D.: Leaf responses to mild droughtstress in natural variants of arabidopsis. Plant physiology, vol. 167, no. 3, pp. 800–816 (2015)
5. Derksen, D., Inglada, J., Michel, J.: Scaling up slic superpixels using a tile-basedapproach. IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 5, pp. 3073–3085 (2019)
6. Elvanidi, A., Zinkernagel, J., Max, J., Katsoulas, N.: Contribution of hyperspec-tral imaging to monitor water content in soilless growing cucumber crop. In: International Symposium on Advanced Technologies and Management for Innovative Greenhouses: GreenSys2019, 1296. pp. 1055–1062 (2019)
7. Gao, Z., Luo, Z., Zhang, W., Lv, Z., Xu, Y.: Deep learning application in plantstress imaging: A review. AgriEngineering, vol. 2, no. 3, pp. 430–446 (2020)
8. Ghosal, S., Blystone, D., Singh, A. K., Ganapathysubramanian, B., Singh, A.,Sarkar, S.: An explainable deep machine vision framework for plant stress phe-notyping. In: Proceedings of the National Academy of Sciences, vol. 115, no. 18, pp. 4613–4618 (2018)
9. Joalland, S., Screpanti, C., Varella, H. V., Reuther, M., Schwind, M., Lang, C.,Walter, A., Liebisch, F.: Aerial and ground based sensing of tolerance to beet cystnematode in sugar beet. Remote Sensing. vol. 10, no. 5, pp. 787 (2018)

10. Katsoulas, N., Elvanidi, A., Ferentinos, K. P., Kacira, M., Bartzanas, T., Kittas, C.: Crop reflectance monitoring as a tool for water stress detection in greenhouses: A review. Biosystems Engineering, vol. 151, pp. 374–398 (2016)
11. Khanna, R., Schmid, L., Walter, A., Nieto, J., Siegwart, R., Liebisch, F.: A spatiotemporal spectral framework for plant stress phenotyping. Plant methods, vol. 15, no. 1, pp. 1–18 (2019)
12. Kirchgessner, N., Liebisch, F., Yu, K., Pfeifer, J., Friedli, M., Hund, A., Walter, A.: The ETH field phenotyping platform fip: A cable-suspended multi-sensor system. Functional Plant Biology, vol. 44, no. 1, pp. 154–168 (2016)
13. Liñan-Vigo, F. N.: Plasticidad fenotípica en plantas de origanum vulgare "ni-gra"(orégano) en respuesta al estrés hídrico en condiciones de invernadero (2018)
14. Luna-Flores, W., Estrada-Medina, H., Jimenez-Osornio, J., Pinzon-Lopez, L.: Effect of water stress on growth and water use efficiency of tree seedlings of threedeciduous species. Terra Latinoamericana, vol. 30, no. 4, pp. 343–353 (2012)
15. Naik, H. S., Zhang, J., Lofquist, A., Assefa, T., Sarkar, S., Ackerman, D., Singh, A., Singh, A. K., Ganapathysubramanian, B.: A real-time phenotyping frame work using machine learning for plant stress severity rating in soybean. Plant methods, vol. 13, no. 1, pp. 1–12 (2017)
16. Neilson, E. H., Edwards, A. M., Blomstedt, C., Berger, B., Møller, B. L., Gleadow, R. M.: Utilization of a high-throughput shoot imaging system to examine the dy-namic phenotypic responses of a c4 cereal crop plant to nitrogen and water defi-ciency over time. Journal of experimental botany, vol. 66, no. 7, pp. 1817–1832 (2015)
17. Ning, Y., Zhang, H., Zhang, Q., Zhang, X.: Rapid identification and quantitative pitmud by near infrared spectroscopy with chemometrics. Vibrational Spectroscopy vol. 110, pp. 103–116 (2020) doi.org/10.1016/j.vibspec.2020.103116
18. Nowosad, J., Stepinski, T.: Generalizing the simple linear iterative clustering (slic) superpixels (2021)
19. Osuna-Coutiño, J. A. D J., Martinez-Carranza, J.: Volumetric structure extraction in a single image. The Visual Computer, Springer (2021) doi: 10.1007/s00371-021-02163-w
20. Pieruschka, R., Schurr, U.: Plant phenotyping: Past, present, and future. Plant Phenomics 2019 (2019) doi: 10.34133/2019/7507131
21. Rist, F., Gabriel, D., Mack, J., Steinhage, V., Topfer, R., Herzog, K.: Combination of an automated 3d field phenotyping workflow and predictive modelling forhigh through put and non-invasive phenotyping of grape bunches. Remote Sensing, vol. 11, no. 24, pp. 2953 (2019)
22. Rodriguez de Cianzio, S., WR, F., Ic, A.: Genotypic evaluation for iron deficiency chlorosisin soybeans by visual scores and chlorophyll concentration (1979), 24. Singh, A.K., Ganapathysubramanian, B., Sarkar, S., Singh, A.: Deep learning forplant stress phenotyping: trends and future perspectives. Trends in plant science, vol. 23, no. 10, pp. 883–898 (2018)
23. Vasseur, F., Bontpart, T., Dauzat, M., Granier, C., Vile, D.: Multivariate genetic analysis of plant responses to water deficit and high temperature revealed con-trasting adaptive strategies. Journal of experimental botany, vol. 65, no. 22, pp. 6457–6469 (2014)
24. Walter, A., Finger, R., Huber, R., Buchmann, N.: Opinion: Smart farming is keyto developing sustainable agriculture. In: Proceedings of the National Academy of Sciences, vol. 114, no. 24, pp. 6148–6150 (2017)
25. Wang, Y., Wang, D., Zhang, G., Wang, J.: Estimating nitrogen status of rice usingthe image segmentation of gr thresholding method. Field Crops Research, vol. 149, pp. 33–39 (2013)

ISSN 1870-4069

3D Convolutional Neural Network for Positron Emission Tomography Image Enhancement

Leandro José Rodríguez Hernández[1],
Humberto de Jesús Ochoa Domínguez[1],
Vianey Guadalupe Cruz Sánchez[1],
Osslan Osiris Vergara Villegas[1], Juan Humberto Sossa Azuela[2]

[1] Universidad Autónoma de Ciudad Juárez,
Instituto de Ingeniería y Tecnología,
Mexico

[2] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{hochoa,vianey.cruz,overgara}@uacj.mx,
hsossa@cic.ipn.mx, al194726@alumnos.uacj.mx,

Abstract. In this work, we propose a 3D convolutional neural network (CNN) for positron emission tomography (PET) image enhancement as an application of the artificial intelligence (AI) in the area of health. Our proposed network manages to increase the number of counts in the PET sinograms, thus, positively influencing the final quality of the reconstructed image. The enhanced sinogram, obtained by the network, is reconstructed using the ordered subset expectation maximization (OSEM) algorithm. The results show that the proposed network is able to increase the PSNR by 6% on average and the contrast almost twice.

Keywords: Positron emission tomography, convolutional neural network, sinogram, image enhancement.

1 Introduction

Industry 4.0 is changing the way medical devices are produced and delivered. Artificial intelligence (AI) has become a key element of this industry and it is knocking down barriers and forcing to assess the way the traditional work is done. In the medical area, the AI trends and solutions are transforming the medical imaging field to improve the diagnosis process [1].

Medical imaging is the set of techniques used to inspect the human body, with the objective of diagnosing, monitoring, or treating medical conditions [2]. Positron emission tomography (PET) is a technique to acquire images representative of the metabolic activity of the body. At the beginning of the PET scan, the patient is injected with a small dose of radioactive material called radiotracer.

For a cancer study, the radiotracer used is fluorodeoxyglucose (F^{18}). This substance decays by a neutrino and a positron (beta+) with a lifetime of about 109 minutes. Each

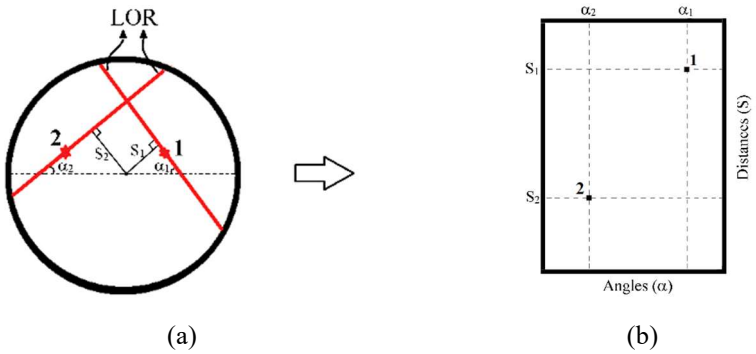


Fig. 1. Representation of (a) two LORs where two and one events have happened respectively and (b) resulting 2D sinogram with the events stored.

positron annihilates with an electron, producing two photons of high energy, traveling in opposite directions. This is known as coincidence or event. Each event is counted upon reaching the scanner's detectors within a time window. The point of annihilation is ideally located on a straight line connecting a pair of detectors.

This line is known as the line of response (LOR) [3]. Reconstruction can be performed using filtered back projection methods or iterative methods. In a PET study, the time is reduced, not all the events are counted as true coincidences, but they can also be random or scattered coincidences. In the last two cases, the coincidence detected by each of the detectors comes from different LORs.

Randoms are one of the main sources of degradation of the image, since they introduce noise, making difficult the quantification. Also, the acquisition time is reduced, resulting in noisy and low resolution images [2, 3]. The Poisson noise affects the image quality negatively, influencing the detectability of lesions and the medical diagnoses. It is the current interest of the PET community to find methods to recover and preserve the important information in the images. In this paper, we present an application of the AI in the health area.

Our proposal consists of a three-dimensional (3D) CNN to improve the PET sinogram, demonstrating that it is feasible and important to retrieve information between slices, but also intra slices. The paper is organized as follows. In section 2, the recent related works are presented. Section 3 presents the main concepts related to the topic and the proposed method. In section 4, we show the results of the experimentation and, finally in Section 5, conclusions are presented.

2 Related Work

A set of techniques have been proposed to solve the problem of poor quality in PET images. Conventional approaches include processing algorithms [4, 5], anatomically guided [6], and magnetic resonance imaging (MRI) guided algorithms with partial volume correction [7]. Although these methods try to minimize noise, loss of spatial resolution is still observed. The algorithms of artificial intelligence have been included

in the area of medical image reconstruction and enhancement. Most of the works focus on the reconstructed images.

They propose to use trained networks with pairs of low resolution and high resolution images [8,9]. The high-resolution images are obtained from an acquisition with a modern ultra-high definition scanner, and degraded to obtain the low-resolution version. Other authors incorporate into the network training, anatomical information obtained from a computed tomography (CT) or MRI scanner [10, 11], arguing that this information is useful for estimating a more robust model and higher quality images.

Recent studies have focused their efforts on improving the sinogram instead of the reconstructed image. For example, in [12], the authors use Monte Carlo simulations and CNN to recover improved 2D sinograms from the low-resolution originals produced by simulated tomographs with large and small crystals.

2.1 The Importance of the Third Dimension

With the increase in computing power, some researchers have explored the possibility of addressing the problem of the low resolution of PET images in a three-dimensional way using deep learning systems.

For example, in [13], a 3D variant of the U-Net network is proposed to improve reconstructed images, reducing noise in PET images of the brain and chest. We think that in the domain of the sinogram is possible to obtain advantages if the sinogram is processed as a volume instead of sequential two-dimensional (2D) slices.

Despite the more common view of sinograms, as 2D slices, however, in PET, the acquired sinogram are volumes. Sinograms are three-dimensional structures, where not only the intra slice information (x, y, axes) is important but also the inter slices information (z-axis). In this work, we propose a 3D CNN to improve the quality of the acquired 3D sinograms.

3 Materials and Methods

This section presents the main methods and concepts related to PET tomography. Terms related to CNNs are also explained, as well as the 3D CNN proposed in this work is detailed.

3.1 PET Sinograms

A 2D sinogram is a matrix whose axes correspond to the angles (α) versus the orthogonal distances (S) of the orthogonal lines from the center of the tomograph to the LOR. Figure 1(b) shows a 2D sinogram with two events from the LORs shown in Figure 1(a).

In this case, on one of the LORs, at an angle (α_1), occurred two events and on the LOR at an angle (α_2) occurred one event. The number of possible angles is determined by the number of crystals per ring and the number of possible distances.

Also, this number depends on the number of crystals, and the transaxial field of view of the scanner [2,3]. Figure 2(a) shows a scanner with the arrow pointing to the axial axis. Figure 2(b) shows the acquisition of 2D sinograms. In this case, the events whose

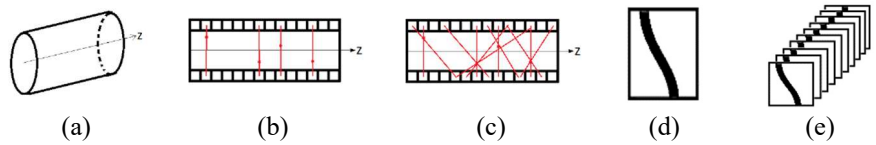


Fig. 2. Events acquisition from (a) a tomograph, (b) acquisition of 2D sinograms (one sinogram per ring detector), (c) acquisition of a 3D sinogram (only one volume), (d) 2D sinogram and (e) 3D sinogram.

LORs occur in the same axial plane of each detector ring are recorded in different 2D slices. Figure 2(c) shows that the events detected by crystals, from different rings, are stored in volumetric files, also called 3D sinograms.

This implies a substantial increase in the size and in the reconstruction time. The quality of reconstructed images is higher, due to the greater amount of information available. All the information detected in a 3D acquisition is stored in a 3D sinogram, where each corresponds to a 2D sinogram. Figures 2(d) and 2(e) shows the representation of 2D and 3D sinograms respectively.

3.2 PET Ordered Subset Expectation Maximization Algorithm

The maximum likelihood expectation-maximization algorithm (MLEM) was introduced in the field of image reconstruction in [14]. A variant of the MLEM method is the ordered subset expectation-maximization algorithm (OSEM) [15]. This method groups the PET scanner detectors into subsets to perform the processing of each subset in iterations, doing one subset at a time. OSEM reduces the reconstruction time relative to MLEM.

3.3 Scanner MicroPET FOCUS 220

We simulated the MicroPET FOCUS 220 preclinical scanner using the Gamos software [16] to perform the experiments. The scanner consists of four detector rings: each ring is made up of 42 detector blocks. Each detector block is composed of a matrix of 12 × 12 LSO crystals with dimensions of 1.5 mm × 1.5 mm × 10.0 mm. Its axial field of view is 7.6 cm and its transaxial field of view is 19 cm [17]. So that, one can acquire a 3D sinogram of size 252 × 287 × 2304.

3.4 Convolutional Filters

The convolution filters are two-dimensional or three-dimensional matrices to perform the convolution operation on the image to extract different characteristics. Typically, they are of size 3 × 3 or 5 × 5. The filter moves through the image from left to right, and from top to bottom, and in the case of 3D, from front to back, advancing a certain number of steps known as a stride.

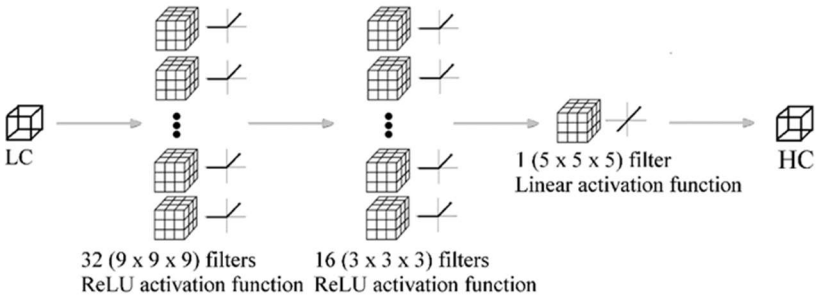


Fig. 3. Proposed 3D CNN.

Table 1. Resulting hyperparameters after tuning the proposed 3D CNN.

Parameter	Value
Loss function	Mean Squared Error (MSE)
Optimization algorithm	Adam
Learning rate	0.0003
Batch size	128
Epochs	200
Size of filters in the input layer	9 × 9 × 9
Size of filters in the mapping layer	3 × 3 × 3
Size of filters in the transpose convolution layer	5 × 5 × 5
Number of filters in the input layer	32
Number of filters in the mapping layer size	16
Number of filters in the transpose convolution layer	1

3.5 Convolutional Neural Networks

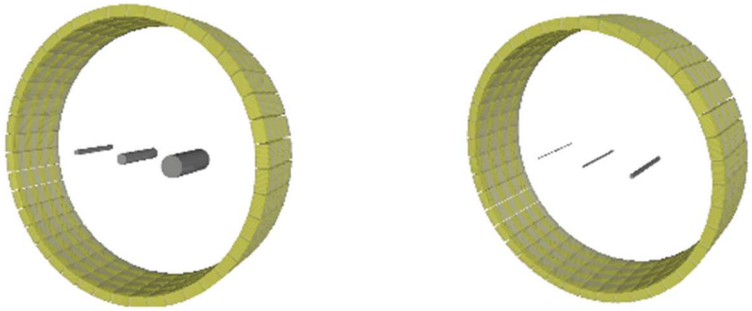
A CNN is a deep learning algorithm that can incorporate an input image, assign importance (weights and learnable biases) to various aspects or objects in the image and differentiate one from another, among other activities.

The preprocessing required in a CNN is much less compared to other classification algorithms. While in primitive methods, the filters are designed by hand, with enough training, CNNs can learn these filters.

Also, a CNN can capture spatial and temporal dependencies in an image through the application of filters.The architecture is better suited to the image data set due to the reduction in the number of parameters involved and the reuse of weights [18].

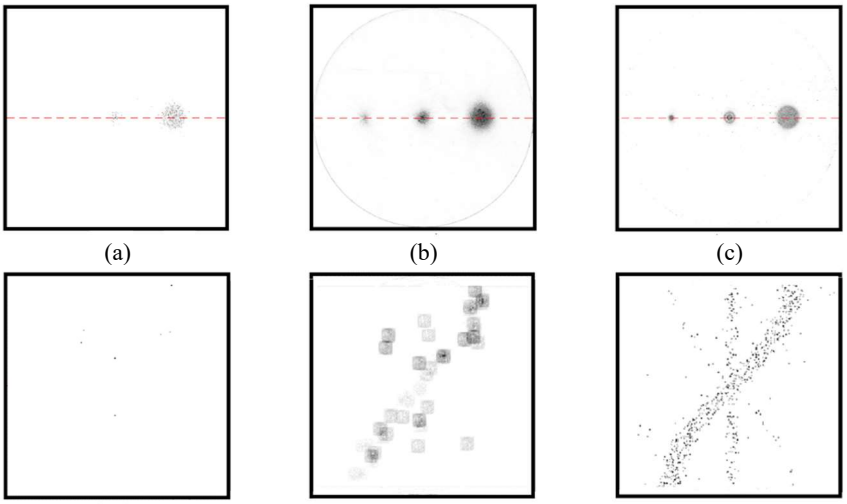
3.6 Deconvolution

Deconvolution is the inverse operation of convolution. It is used to recover data degraded by a physical process modeled as a convolution. If the degraded signal and



(a) (b)

Fig. 4. Simulated phantom of dimensions (a) 5 mm, 10 mm and 20 mm in diameter and 6 cm in length and (b) 1 mm, 2 mm, and 5 mm in diameter and 6 cm in length.



(a) (b) (c)

(d) (e) (f)

Fig. 5. MIPs of the transversal view of the simulated phantom with three cylinders of 5 mm, 10 mm and 20 mm in diameter. (a) LC (PSNR=26.33 dB, C=1.1680), (b) enhanced (PSNR=32.01 dB, C=2.5014) and (c) ground truth phantoms. Middle slice from the 3D (d) LC sinogram, (e) recovered sinogram from the proposed 3D CNN and (f) ground truth sinogram.

the system are known, it is possible to find the original signal by a deconvolution operation [19].

3.7 Dataset

High counts (HC) and low counts (LC) sinograms are generated to train the network. Five phantoms were simulated for the generation of the sinograms, containing spheres with diameters ranging from 0.5mm to 5mm. The spheres are randomly distributed within the simulated phantoms, so that the sinograms obtained contain enough useful

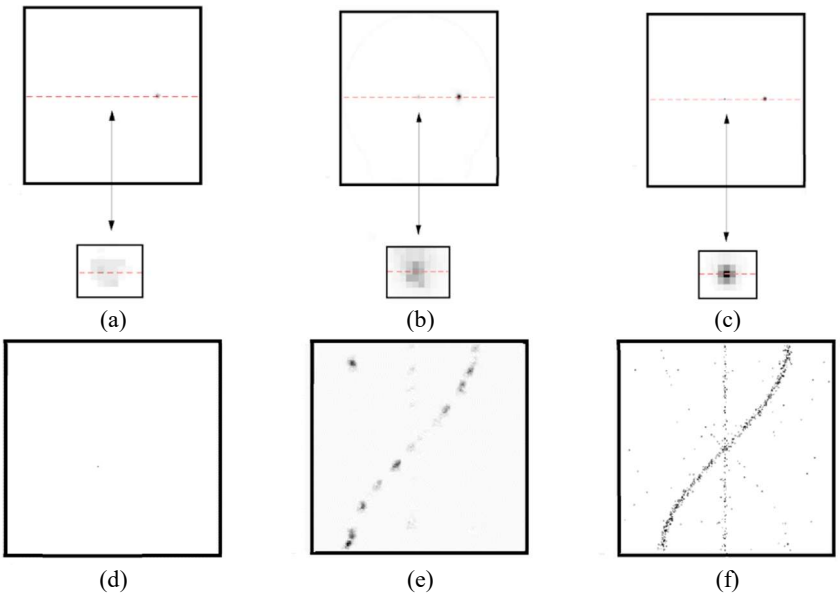


Fig. 6. MIPs of the transversal view of the simulated phantom with three cylinders of 1 mm, 2 mm and 5 mm in diameter. (a) LC (PSNR=25.12 dB, C=1.1521), (b) enhanced (PSNR=27.23 dB, C=2.4072) and (c) ground truth phantoms. Middle slice from the 3D (d) LC sinogram, (e) recovered sinogram from the proposed 3D CNN and (f) ground truth sinogram.

information for training. The LC sinogram is generated with 10 million events, of which only effective PET events are, about 5%.

One hundred million events were generated in the phantom to obtain the HC sinograms of which only 5% are true events. For network training, we extract 3D patches of size 32 × 32 × 32, selected randomly from the pairs of LC and HC sinograms. Our training set contains 15250 LC-HC patch pairs, while our validation set consists of 5000 patch pairs.

3.8 Proposed 3D CNN for PET Sinogram Enhancement

Figure 3 shows the diagram of the 3D CNN proposed to improve low resolution sinograms. The network has three layers. The input layer extracts the characteristics of the low-resolution sinograms, the mapping layer performs the mapping between the low resolution and high-resolution features, and the third layer or transpose convolution (deconvolution) layer performs the final reconstruction.

The input layer consists of 32 filters of size 9 × 9 × 9, with the ReLU [20]. The mapping layer has 16 filters of 3 × 3 × 3, with ReLU. The third layer has one filter of 5 × 5 × 5, with a linear activation function. Table 1 shows the hyperparameters adjusted by using the grid search method for the proposed 3D CNN. The proposed network has three layers only. Therefore, it is feasible and more efficient to tune the hyperparameters by the grid search method [21, 22].

4 Experiments

In this section, we present the methodology of evaluation and discuss the experimental results. We simulated six rods of 1 mm, 2 mm, 5 mm, 10 mm and 20 mm using the scanner MicroPET FOCUS 220. To examine the performance, we utilized the metrics peak signal to noise ratio (PSNR) and contrast (C) according to [23].

In our case, contrast indicates the difference between the lesions and the background in the image. If the difference is large, the lesions are more distinguishable.

4.1 Methodology of Evaluation

The maximum intensity projection (MIP) of the transversal views was used to obtain the quantitative results. A series of case studies were tested to verify the efficiency of the proposed method. In each experiment, the original low quality phantom was simulated with 500 thousand events and its corresponding ground-truth with 10 million events. For the simulation of the test phantoms, the Gamos software [16] was used. The OSEM algorithm [15] was used for the reconstruction.

4.2 Experimental Results

Figure 4(a) shows a phantom with three cylinders of 5 mm, 10 mm, and 20 mm in diameter and 6 cm in length, respectively. Figure 4(b) shows a phantom with three cylinders of 1 mm, 2 mm, and 5 mm in diameter and 6 cm in length. Both phantoms filled with F^{18} were used to acquire 3D sinograms for testing. Figure 5 shows the MIPs of the reconstructed images with OSEM from three cylinders of 5 mm, 10 mm, and 20 mm in diameter and their corresponding sinograms.

Figure 6 shows the MIPs of the reconstructed images with OSEM from three cylinders of 1 mm, 2 mm, and 5 mm in diameter and their corresponding sinograms. The proposed network increases the number of counts and the contrast, which positively affects the visualization. For example, Figure 7 (a) shows the MIPs of Figure 5. In this case, the proposed CNN recovers the 5mm rod. Figure 7 (b) shows the MIPs of Figure 6. In this case, the proposed CNN also recovers the 1mm rod.

The results obtained in the experiments allow us to affirm that the proposed method applied to the PET sinograms, in three dimensions, permits to increase the contrast of the rods in the reconstructed images.

A better definition of the limits between the area containing the radiotracer and the background area is achieved. Figures 5(a to c) and 6(a to c) visually show that when the LC sinograms are processed by our method, valuable information is recovered when reconstructed.

For example, in Figure 6(b), it is possible to observe how the proposed method recovers the intermediate lesion of 2mm in diameter, when it was originally almost invisible Figure 6(a). Table 2 compares the PSNR and contrast after recovering the low count phantom and the enhanced phantom. Notice that our network results (bold values) increase the PSNR by 6% on average and the contrast almost twice. Figures 5(d to f) and 6(d to f) clearly shows how the proposed method manages to recover effective counts in the sinograms, while the LC sinograms (Figures 5(d) and 6(d)) contain little

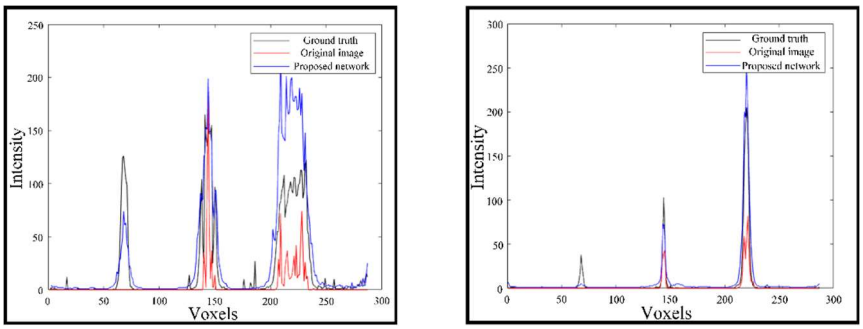


Fig. 7. Horizontal profiles of the (a) Figure 5 for the MIP of the phantom of dimensions 5 mm, 10 mm, and 20 mm in diameter and (b) Figure 6 for the MIP of the phantom of dimensions 1 mm, 2 mm, and 5 mm in diameter.

Table 2. Summary of results.

		LC phantom	Enhanced phantom
Case study 1	PSNR	26.33 dB	**32.01 dB**
	Contrast	1.1680	**2.5014**
Case study 2	PSNR	25.12 dB	**27.23 dB**
	Contrast	1.1521	**2.4072**

Note: Bold values indicate the best results.

information, after being processed by our method, useful information is recovered for reconstruction (Figures 5(e) and 6(e)).

5 Conclusions

In this article, we presented an application of the AI in the health area. A 3D CNN was proposed to enhance sinograms of PET images.

Sinograms with a high number of counts and their corresponding sinograms with few counts were generated 32 × 32 ×32 patches were extracted from both pairs of sinograms to form the training sets. An analysis of the profiles obtained from the MIP of the reconstructed images is carried out.

The results show that the proposed network can increase the number of counts in the sinogram, which positively influences the quality of the images. Our method demonstrates the importance of treating the sinogram as a three-dimensional structure to consider the intra-slice information.

The main advantages are the improvement of the 3D sinograms and the 3D CNN architecture with three layers only. However, the main drawback is that the method needs to be trained first, so that it is computationally expensive and time-consuming.

Acknowledgments. L. J. Rodríguez thanks the UACJ for the support provided and the CONACYT for the scholarship granted to carry out his doctoral studies. H. Sossa thanks the Instituto Politécnico Nacional for the support under projects SIP 20210788, and CONACYT under projects 65 (Fronteras de la Ciencia) and 6005 (FORDECYTPRONACES).

References

1. Al-Jaroodi, J., Mohamed, N., Abukhousa, E.: Health 4.0: On the way to realizing the healthcare of the future. IEEE Access, vol. 8, pp. 211189–211210 (2020) doi: 10.1109/ACCESS.2020.3038858
2. Saha, G. B.: Pet scanning systems. In: Basics of PET Imaging. Springer-Verlag New York (2010)
3. Cherry, S., Dahlbom, M.: PET Physics, Instrumentation, and Scanners. Springer-Verlag, New York (2006)
4. Arabi, H., Zaidi, H.: Improvement of image quality in pet using post-reconstruction hybrid spatial-frequency domain filtering. Physics in Medicine & Biology, vol. 63**,** pp. 215010 (2018) doi: 10.1088/1361–6560/aae573
5. Arabi, H., Zaidi, H.: Spatially guided nonlocal mean approach for denoising of pet images. Medical Physics, vol. 47, pp. 1656–1669 (2020) doi: 10.1002/mp.14024
6. Chan, C., Fulton, R., Barnett, R., Feng, D., Meikle, S.: Postreconstruction nonlocal means filtering of whole-body pet with an anatomical prior. IEEE Transactions on Medical Imaging, vol. 33**,** pp. 636–650 (2014) doi: 10.1109/TMI.2013.2292881
7. Yan, J., Lim, J., Townsend, D.: MRI-guided brain pet image filtering and partial volume correction. Physics in Medicine & Biology, vol. 60, no. 3, pp. 961–976 (2015)
8. Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., Sánchez, C. I.: A survey on deep learning in medical image analysis. Medical Image Analysis, vol. 42, pp. 60–88 (2017) doi: 10.1016/j.media.2017.07.005
9. Chen, K., Gong, E., de Carvalho, F., Xu, J., Boumis, A., Khalighi, M., Poston, K. L., Sha, S. J., Greicius, M. D., Mormino, E. Pauly, J. M., Srinivas, S., Zaharchuk G.: Ultra–Low-Dose (18) F-Florbetaben Amyloid PET Imaging Using Deep Learning with Multi-Contrast MRI Inputs. Radiology, vol. 290, 649–656 (2019) doi: 10.1148/radiol.2018180940
10. Xu, J., Gong, E., Pauly, J., Zaharchuk, G.: 200x Low-dose PET reconstruction using deep learning. ARXIV (2017) eprint arXiv:1712.04119
11. Liu, C., Qi, J.: Higher SNR PET image prediction using a deep learning model and MRI image. Physics in Medicine & Biology, vol. 64, no. 2019, pp. 115004, doi: 10.1088/1361–6560/ab0dc0
12. Hong, X., Zan, Y., Weng, F., Tao, W., Peng, Q., Huang, Q.: Enhancing the image quality via transferred deep residual learning of coarse pet sinograms. IEEE Transactions on Medical Imaging, vol. 37, pp. 2322–2332 (2018)
13. Lu, W., Onofrey, J. A., Lu, Y., Shi, L., Ma, T., Liu, Y., Liu, C.: An investigation of quantitative accuracy for deep learning based denoising in oncological PET. Physics in Medicine & Biology, vol. 64, pp. 5019 (2019)
14. Shepp, L., Vardi, Y.: Maximum likelihood reconstruction for emission tomography. IEEE Transactions on Medical Imaging, vol. 1, pp. 113–122 (1982)
15. Hudson, H., Larkin, R.: Accelerated image reconstruction using ordered subsets of projection data. IEEE Transactions on Medical Imaging, vol. 13, pp. 601–609 (1994)
16. Arce, P., Lagares, J., Harkness, L., Pérez-Astudillo, D., Cañadas, M., Rato, P., de Prado, M., Abreu, Y., de Lorenzo, G., Kolstein, M., Díaz, A.: Gamos: A framework to do Geant4 simulations in different physics fields with a user-friendly interface. Nuclear Instruments and Methods in Physics, vol. 735, pp. 304–313 (2014)
17. Tai, Y., Ruangma, A., Rowland, D., Siegel, S., Newport, D., Chow, P., Laforest, R.: Performance evaluation of the microPET focus: a third-generation microPET scanner dedicated to animal imaging. J Nucl Med., vol. 46, pp. 455–463 (2005)
18. Socher, R., Huval, B., Bath, B., Manning, C. D., Ng, A.: Convolutional-recursive deep learning for 3D object classification. Advances in Neural Information Processing Systems, vol. 25 (2012)

19. Zeiler, M., Krishnan, D., Taylor, G., Fergus, R.: Deconvolutional networks. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2528–2535 (2010) doi: 10.1109/CVPR.2010.5539957
20. Nair, V., Hinton, G.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of ICML. vol. 27, pp. 807–814 (2010)
21. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 437–478 (2012) doi: 10.1007/978-3-642-35289-8_26
22. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research, vol. 13, pp. 281–305 (2012)
23. Kennedy, J., Israel, O., Frenkel, A., Bar-Shalom, R., Azhari, H.: Super-resolution in PET imaging. IEEE Transactions on Medical Imaging, vol. 25, pp. 137–147 (2006) doi: 10.1109/TMI.2005.861705

ISSN 1870-4069

Recognition of Stereotypical Motor Movements in Children with Autism Spectrum Disorder Using a ConvLSTM Network

Magdiel García-Juárez[1], José Anibal Arias-Aguilar[1], Alberto Elías Petrilli-Barceló[2]

[1] Universidad Tecnológica de la Mixteca, División de Estudios de Postgrado, Oaxaca, Mexico

[2] Tokyo University of Science, Faculty of Science and Technology, Japan

ps2017280001@ndikandi.utm.mx, anibal@mixteco.utm.mx, petrilli@rs.tus.ac.jp

Abstract. Stereotypical motor movements (SMM) are behaviors typically found in children with autism spectrum disorder (ASM). They consist of repetitive movements such as hand flapping, body rocking, and finger flicking. Automatically monitoring and recognizing these movements in a way that is reliable and efficient over time could provide important information that would improve our understanding and contribute to an intervention strategy around a central ASM symptom. This paper proposes a model based on deep-learning techniques for automatically classifying video-recorded SMMs, which are present in children with ASM. For obtaining spatial information, the YOLOv2 object-detection architecture is used, in which each video frame is processed and converted into a new abstract representation of 13 x 13 x 1024 descriptors, which are then input into a Recurrent Convolutional Network to find temporal information and classify the actions. The proposed architecture is trained by the SSDB skill-assessment dataset, composed of low-quality videos in uncontrolled environments in which children with ASM are performing SMMs. To compensate for the disadvantage of having a low number of examples, we propose doing a pretraining with the HMDB51 dataset, which has actions with a movement dynamic similar to the proposed actions.

Keywords: Autism, deep learning, HAR.

1 Introduction

Autism spectrum disorder (ASM) is a neurodevelopmental disorder characterized by diagnostic criteria that include significant problems in communication and social interaction; limited patterns of behavior, interests, or activities; and repetitive movements called stereotypical motor movements (SMM) [1, 2]. The most frequent

SMMs include repetitive movements such as body rocking and complex hand-and-finger movements [3].

The skill-assessment process of children with ASM generally involves asking them to perform certain actions by giving them a set of instructions and monitoring their responses as they happen. Interviews with specialists, behavioral observations, and parental reports are needed.

This process involves carrying out work-intensive tasks such as recording the observations of their action responses to a set of stimuli over long periods of time [4]. The automatic, reliable, and efficient detection and monitoring of SMMs over time could provide benefits not only for diagnosing autistic children, but also for a better understanding of and accounting for diverse elements in developing an intervention strategy for a central ASM symptom [5, 6].

To confront this challenge, deep learning (DL) techniques for human action recognition (HAR) by processing video frames allow for the automatization of many SSM-monitoring tasks in ASM therapy, making the diagnosis of autism easier. In this paper, we propose a model based on deep learning techniques for the automatic detection of SMMs in children with ASM (see Fig. 1).

We evaluate their performance using the open-access dataset SSDB [7], which is composed of videos compiled from public-access websites posted by parents or caregivers that were filmed in unregulated settings, and thus may have obstructions, poor lighting, and a smaller number of each example type. To compensate for the fact that the lack of data may affect the model is learning, we propose pretraining the network with the HMDB51 dataset [8], from which we had to choose action categories that had similar movements to the SSDB categories.

One contribution of this paper is its evaluation of the performance of the YOLOv2 architecture, which was originally proposed to detect objects, in its adaptation to the task of the recognition of actions performed over time, using the cascade setting with a Convolutional Long Short-Term Memory (ConvLSTM) Network.

Using YOLOv2, each video frame is converted into a new, more complex representation of descriptors with dimensions of 13 x 13 x 1024, which provides spatial information. These are then input into the ConvLSTM, which will allow temporal information to be extracted for the classification of actions with a Fully Connected (FC) layer. We propose using a ConvLSTM to avoid the problem of flattening input data and losing spatial information that could be important for classifying actions.

2 Related Papers

Video human action recognition is one of the main challenges for the field of computer vision. HAR has had an important impact on applications for children with ASM, as indicated in the paper by Zunino et al., in which hand movements are analyzed while performing the actions of picking up, placing, pouring out, and passing a bottle [9]. The tests showed that children can be classified into groups based on whether they had ASM or not. Using the VGG-16 network in cascade with an LSTM, an accuracy rate of 82% is achieved.

Using the same dataset, Tian et al. [10] and Sun et al. [11] report an improvement in accuracy, with 87.17% and 95.2%, respectively. To detect typical and stereotypical

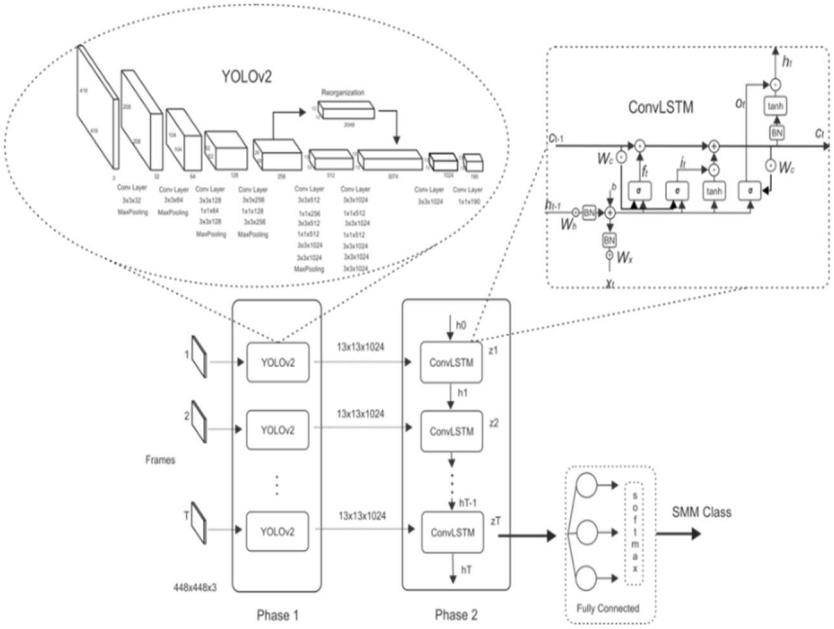


Fig. 1. Model for recognizing SMMs in children with ASM.

Table 1. Pairing of actions from the SSDB and HMDB51 datasets.

SSDB			**HMDB51**		
Action	Train	Eval	Action	Train	Eval
Arm Flapping	45	11	Clap	76	20
Head Banging	23	6	Brush Hair	72	20
Spinning	28	5	Turn	191	48

actions in children with ASM, Silva et al. [12] use the Intel RealSense camera and the SDK Nuitrack to detect and extract the coordinates of the body's joints.

A CNN classifies the different behavioral patterns and obtains an average accuracy of 92.6 percent in the test data. In their paper, Pandey et al. address the problem of action recognition on SSDB dataset classes and a real-world autism dataset.

For the training, the guided weak supervision (GWS) technique was proposed, in which dataset classes are matched through output vectors using the posterior likelihood maximization principle [13].

The YOLOv2 network has shown a high level of performance in the detection of objects, which is why it is a reliable architecture for the task of extracting spatial information. In connecting it in cascade to ConvLSTM, as is shown in Fig. 1, a temporal information-processing model is added.

This collaborative architecture for spatiotemporal processing should provide an efficient model for the task of action recognition for children with ASM. Because few

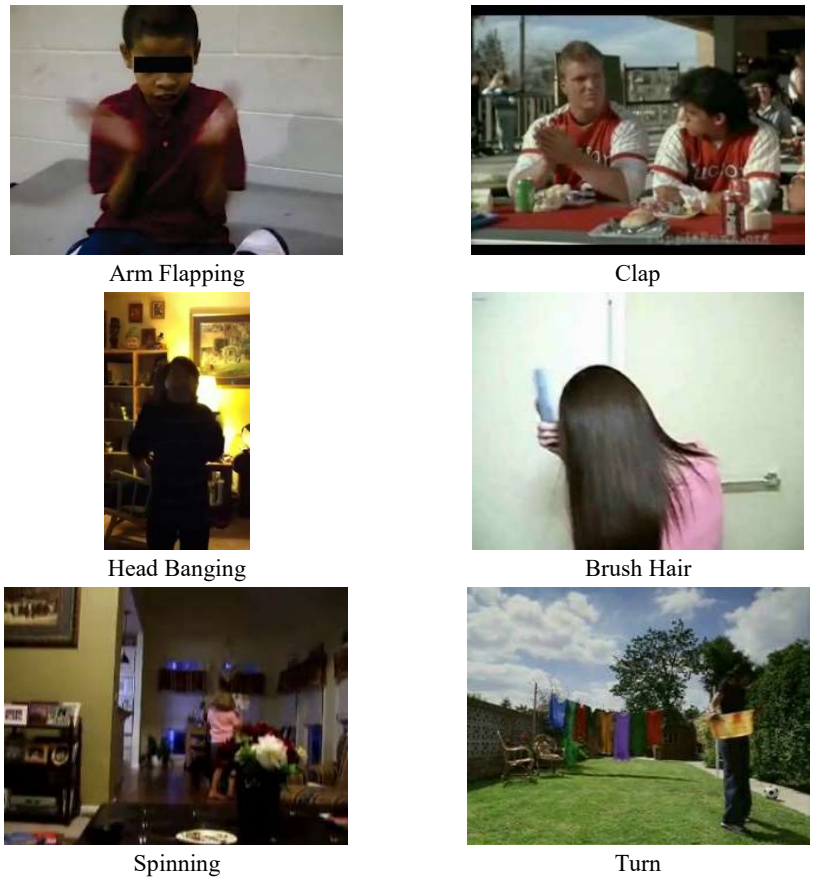


Fig. 2. Examples of actions in the SSDB (left) and HMDB51 (right) datasets.

Table 2. Hyperparameters for the ConvLSTM.

Hyperparameter	**Value**
Dropout	25%
Learning rate	10-4
Loss function	Cross entropy
Timestep	20 units
Optimizer	Gradient descent

data are available for training, we have here adopted the methodology proposed by Pandey to compensate for this lack.

Pandey's methodology consists in pretraining the model using an alternate dataset with actions that are similar to the target-action movements. With this pairing of actions, the goal is for the network to learn from actions as similar as possible to the target actions during the pretraining.

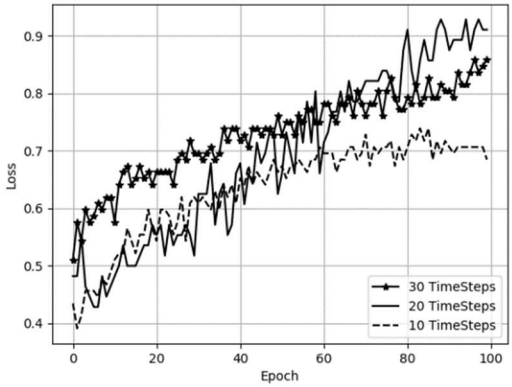


Fig. 3. Accuracy tests with a configuration, in which $timeSteps = [30, 20, 10]$.

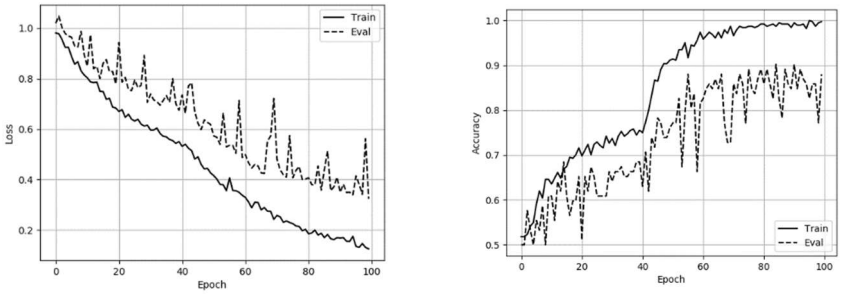


Fig. 4. ConvLSTM-FC's error (left) and accuracy (right) with the HMDB51 dataset.

3 Background

In this section, we discuss the general features of the architectures used for this paper.

3.1 YOLOv2

YOLOv2 addresses object detection as a regression problem in which a single neural network predicts each object's bounding box (BB), and class probabilities are obtained directly from the image to be evaluated [14]. Fig. 1 shows a general diagram of YOLOv2's architecture within the dotted oval.

It is chiefly made up of convolutional layers with batch normalization, which help regularize the model. The input consists of images whose dimensions are 416 x 416, which are divided into *S* x *S* cells, where every cell predicts *B* bounding boxes, which define the frame containing an object, and *C* class probabilities for every BB. Thus, the output vector size for each image is defined as:

$$S \times S \times (B \times (5 + C)). \quad (1)$$

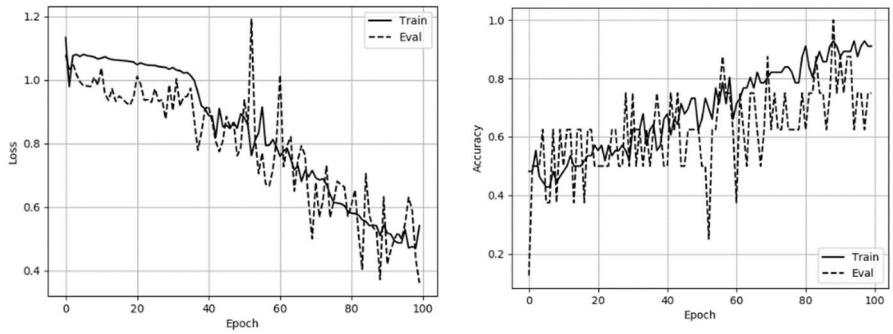


Fig. 5. ConvLSTM-FC error (left) and accuracy (right) with the SSDB dataset.

Table 3. Loss and accuracy in the ConvLSTM-FC training with the HMDB51 and SSDB datasets.

	Loss		**Accuracy**	
	Train	Eval	Train	Eval
HMDB51	0.1	0.35	98 %	88 %
SSDB	0.5	0.45	90%	85 %

The previous version does not have a location-prediction restriction, which makes the early training iterations unstable. For YOLOv2, an Anchor Box (AB) approach was adopted. Anchor Boxes consist of a predefined set of BBs that are better adjusted to the desired objects and that are calculated by the K-means clustering method.

Instead of predicting the BB coordinates relative to the entire image, coordinates are predicted relative to the location of the cell that contains the bounding box. This limits values to between 0 and 1, for which reason the σ(.) logistic activation function is used to restrict the network predictions to this range. In this way, the final BBs are recovered through:

$$b_x = \sigma(t_x) + c_x, \tag{2}$$

$$b_y = \sigma(t_y) + c_y, \tag{3}$$

$$b_w = p_w e^{t_w}, \tag{4}$$

$$b_h = p_h e^{t_h}, \tag{5}$$

$$P(\text{Object}) \times IoU(b, \text{Object}) = \sigma(t_0), \tag{6}$$

where the terms $t_{x,y}$ are the normalized BB center predictions with respect to the cell that contains it, $t_{w,h}$ are the normalized width and height predictions with respect to an AB, $c_{x,y}$ represent the coordinates of the cell that contains the upper left corner of the prediction, and $p_{w,h}$ are the AB's dimensions.

Finally, t_0 is the confidence of having found an object, which is defined as the probability that an object exists multiplied by the Intersection over the Union of the object's BB against the label.

Table 4. Model comparison of HAR performance with the SSDB dataset.

Model	Accuracy
ECO [16]	80.1 %
R(2+1)D [17]	88.3 %
YOLOv2-ConvLSTM	**90 %**
TSM [18]	90.5 %
I3D+GWS+DR [13]	95.7 %

3.2 LSTM Convolutional Networks

Recurrent Neural Networks (RNN) are a type of artificial neural network that, because of their architecture, allow for sequential-data processing. An RNN can be abstracted as though it had multiple copies of itself, with each copy processing the input data in an instant of time *t* and sharing the information with its successor through a cycle called time step (*timeStep*), which is repeated N times. The RNN model's output equation in time step *t* is described as:

$$h_t = g(W_{xh}x_t + W_{hh}h_{t-1} + b_h), \tag{7}$$

$$z_t = g(W_{hz}h_t + b_z), \tag{8}$$

where $h_t \in R^N$ is the hidden state with N hidden units, determined by the value input in network x_t in time step *t* and the previous state's vector h_{t-1} in time step $t-1$. The network's response to input x_t in time step t is defined as z_t.

The matrices W correspond to the network's weights and b is a bias vector. The nonlinear activation function $g(.)$ is commonly defined as sigmoid or hyperbolic tangent.

The LSTM networks improve upon the RNN and avoid the vanishing gradient problem, thanks to the fact that they are able to remember more information in configurations with long periods of time. Generally, LSTM networks adapt well to applications where input data are one-dimensional vectors.

Nevertheless, in applications where the data are *n-dimensional,* they must be flattened, resulting in the loss of information on spatial correlations during the repetitions. This problem can be addressed if we use LSTM (ConvLSTM) convolutional networks [15]. The representation of the ConvLSTM in a time step *t* given an input x_t and the hidden state h_{t-1} is given by:

$$i_t = \sigma(W_{xi} \otimes x_t + W_{hi} \otimes h_{t-1} + W_{ci} \odot c_{t-1} + b_i), \tag{9}$$

$$f_t = \sigma(W_{xf} \otimes x_t + W_{hf} \otimes h_{t-1} + W_{ci} \odot c_{t-1} + b_f), \tag{10}$$

$$o_t = \sigma(W_{x0} \otimes x_t + W_{hc} \otimes h_{t-1} + W_{co} \odot c_t + b_o), \tag{11}$$

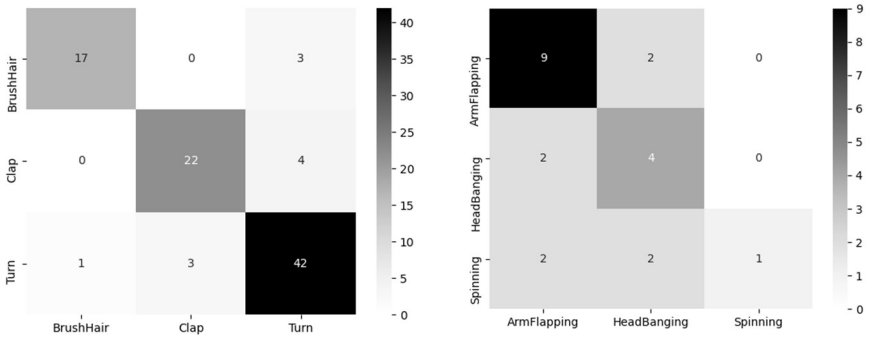


Fig. 6. Confusion matrix with the HMDB51 (left) and SSDB (right) validation data.

$$g_t = tanh(W_{xc} \otimes x_t + W_{hc} \otimes h_{t-1} + b_c), \tag{12}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t, \tag{13}$$

$$h_t = o_t \odot \tanh(c_t). \tag{14}$$

In addition to unit h_t, the ConvLSTM includes the following: an input gate i_t, a gate for forgetting information from earlier states f_t, an output gate o_t, an input modulation gate g_t and a memory cell c_t, which stores information from previous states.

The output gate o_t learns how much is transferred from the memory cell to the hidden state. The symbol $\otimes$ represents the convolutional operation, $\odot$ is the vector product, and $\sigma(.)$ represents the activation function. In the dotted box in Fig. 1, a representation of the basic ConvLSTM unit is shown.

4 Proposed Method

The model proposed for the automatic recognition of the actions of children with ASM by video makes use of DL techniques. A diagram showing the elements that make up the model can be seen in Fig. 1.

The YOLOv2 network in cascade is used with the ConvLSTM-FC network to find a function that assigns input video $V_i = \{v_1, v_2, \dots, v_T\}$ with a length of T frames to the corresponding label $Y \epsilon R^C$, where C is the number of output classes.

Spatial information is sought in phase 1, and thus each frame of video sequence V_{th} is processed using YOLOv 2 to obtain a new representation $X_{th} = \{x_1, x_2, \dots, x_T\}$ corresponding to the features map of the penultimate layer, and having a dimension of 13 x 13 x 1024.

This new representation is used in phase 2 during the training of the ConvLSTM, during which the images' time order and long-range dependencies in the modelling of human actions are recorded. The ConvLSTM output can be defined as $h_{cLSTM} = ConvLSTM(x_{th})$, and the classification result in the FC layer output is given as $y_{th} = softmax(h_{cLSTM})$.

5 Training

The ConvLSTM training is done using the HMDB51 dataset for 100 epochs, and then using the SSDB dataset for the same number of epochs. The mechanism for pairing the HMDB51 and SSDB classes consists in choosing the classes in which the actions are visually similar to the target actions. Table 1 shows the pairing and number of elements per class for training and evaluation. The image resolution varies in each example, with a maximum of 320 x 240 pixels.

In some SSDB videos, it happens that an action is repeated at different moments in the video, or an action might appear that does not correspond to its label. To stop the model from receiving videos in which actions appear that do not belong to the label, the video fragment in which such an action appears was cut out.

In this way, we were able to increase the number of examples from 75 to 113. Some examples of actions from both of the datasets that were used are shown below in Fig. 2, in which very dynamic settings, as well as poor lighting, can be seen.

Due to limitations in computational resources, the new video-frame representations are done offline.Each video of the datasets is processed by the YOLOv2 architecture with the model's original weights, and their new representation is recovered from the penultimate layer of the network, having a dimension of 13 x 13 x 1024.

These new representations are used for training the ConvLSTM-FC network. Table 2 shows the ConvLSTM-FC hyperparameters used during the training.

The computer used has an Intel Xeon 3.5 GHz x 8 processor, 48 GB of RAM memory, and two GeForce RTX 2080 Ti graphics cards with 11 GB of memory each. The open-access TensorFlow v1.6 libraries and the Python 3.0 programming language are run in the Ubuntu 18 operating system.

6 Results

To determine the best option for the timeSteps value in the ConvLSTM network, different tests were run to evaluate the best configuration. Fig. 3 shows the model's performance with a configuration of 30, 20, and 10.

One can see that with a configuration in which *timeSteps*=20, the network performs best. For both datasets, the cost and accuracy functions are monitored during the ConvLSTM-FC network training. Fig. 4 shows the action-classification performance with the HMDB51 dataset, taking into account that only the actions mentioned in Table 1 were used.

Once the network has learned to classify actions that are similar to the target actions, training the network with the SSDB dataset begins in order to refine the knowledge. Fig. 5 shows the network's performance during the training, in which we see less learning stability, especially in the validation data.

The results shown for the cost and accuracy functions in the figures above are summarized in Table 3. In the case of training with the SSDB dataset, the model shows a better performance and is slightly more stable between epochs 90–95. For this reason, the number of epochs is used as a point of reference for evaluating the model.

One way of representing the model's classification performance is by using a confusion matrix. Fig. 6 shows the results with the HMDB51 and SSDB evaluation

data. Table 4 shows a comparison of different models for HAR in their performance with the SSDB dataset. Results with training data are taken into account for this comparison.

7 Conclusions

For this study, a model based on deep learning techniques was implemented for the classification of the actions of children with ASM by video. The experimental tests show that the model made up of the YOLOv2 network in cascade with the ConvLSTM obtains a result within the acceptable range for the state of the art.

Using video frames processed by the YOLOv2 architecture helps the ConvLSTM-FC network in its task of classifying SMM actions. In spite of the fact that the model commits a higher rate of errors with SSDB than with HMDB51, it shows a favorable response in accuracy both with training and evaluation data.

Upon comparing the results obtained with other HAR models using the SSDB dataset, we see that the model proposed here maintains an acceptable range for the state of the art. For future research, we suggest using this model and modifying the cost function for the task of human action localization, which consists of determining the moment in the video in which a human action begins or ends.

For this problem, we suggest considering not just confidence and class probabilities, but also the percentage of the action performed in each frame and considering a new labelling, as well as the incorporation of attention mechanisms in the video frames.

References

1. Kossyvaki, L.: Adult interactive style intervention and participatory research designs in autism: Bridging the gap between academic research and practice. Routledge (2017) doi: 10.4324/9781315719375
2. Kang, J., Han, X., Song, J., Niu, Z., Li, X.: The identification of children with autism spectrum disorder by SVM approach on EEG and eye-tracking data. Computers in biology and medicine. vol. 120 (2020) doi: 10.1016/j.compbiomed .2020.103722
3. Loftin, R. L., Odom, S. L., Lantz, J. F.: Social interaction and repetitive motor behaviors. Journal of Autism and Developmental Disorders, vol. 38, pp. 1124–1135 (2008) doi: 10.1007/s10803-007-0499-5
4. Costa, A. P., Charpiot, L., Lera, F. R., Ziafati, P., Nazarikhorram, A., van Der Torre, L., Stffgen, G.: More attention and less repetitive and stereotyped behaviors using a robot with children with autism. In: Proceedings of 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), IEEE pp. 534–539 (2018) doi: 10.1109/ROMAN.2018.8525747
5. Sadouk, L., Gadi, T., Essoufi, E. H.: A novel deep learning approach for recognizing stereotypical motor movements within and across subjects on the autism spectrum disorder. Computational intelligence and neuroscience, vol. 2018 (2018) doi: 10.1155/2018/ 7186762
6. Groekathöfer, U., Manyakov, N. V., Mihajlovic, V., Pandina, G., Skalkin, A., Ness, S., Bangerter, A., Goodwin, M. S.: Automated detection of stereotypical motor movements in autism spectrum disorder using recurrence quantification analysis. Frontiers in neuroinformatics, vol. 11, no. 9 (2017)

7. Rajagopalan, S., Dhall, A., Goecke, R.: Self-stimulatory behaviours in the wild for autism diagnosis. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 755–761 (2013)
8. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision, IEEE pp. 2556–2563 (2011) doi: 10.1109/ICCV.2011.6126543
9. Zunino, A., Morerio, P., Cavallo, A., Ansuini, C., Podda, J., Battaglia, F., Veneselli, E., Becchio, C., Murino, V.: Video gesture analysis for autism spectrum disorder detection. In: Proceedings of 24th International Conference on Pattern Recognition (ICPR), IEEE pp. 3421–3426 (2018)
10. Tian, Y., Min, X., Zhai, G., Gao, Z.: Video-based early ASD detection via temporal pyramid networks. In: Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pp. 272–277 (2019) doi: 10.1109/ICME.20 19.00055
11. Sun, K., Li, L., He, N., Zhu, J.: Spatial attentional bilinear 3D convolutional network for video-based autism spectrum disorder detection. In: Proceeding of ICASSP 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3387–3391 (2020) doi: 10.1109/ICASSP40776.20 20.9054641
12. Silva, V., Soares, F., Esteves, J. S., Vercelli, G.: Human action recognition using an image-based temporal and spatial representation. In: Proceedings of 12th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT), IEEE, pp. 41–46 (2020) doi: 10.1109/ICU MT51630. 2020.9222408
13. Pandey, P., Prathosh, A., Kohli, M., Pritchard, J.: Guided weak supervision for action recognition with scarce data to assess skills of children with autism. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 463–470 (2020)
14. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271 (2017)
15. Xingjian, S., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., Woo, W. C.: Convolutional LSTM Network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, pp. 802–810 (2015)
16. Zolfaghari, M., Singh, K., Brox, T.: Eco: Efficient convolutional network for online video understanding. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 695–712 (2018)
17. Ghadiyaram, D., Tran, D., Mahajan, D.: Large-scale weakly-supervised pretraining for video action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12046–12055 (2019)
18. Lin, J., Gan, C., Han, S.: TSM: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7083–7093 (2019)

ISSN 1870-4069

Fatal Cyclist-Car Accidents at Intersections: An Analysis from the Guadalajara Metropolitan Area

Ramon A. Briseño[1], Rocio Maciel Arellano[2], Edgar Cossio[3], Víctor M. Larios[2], Raul J. Beltrán[1], José Antonio Orizaga T.[1],

[1] Universidad de Guadalajara,
Centro Universitario de Ciencias Económico Administrativas,
Doctorado en Tecnologías de Información,
Mexico

[2] Universidad de Guadalajara,
Centro Universitario de Ciencias Económico Administrativas,
Centro de Innovación en Ciudades Inteligentes,
Mexico

[3] Instituto de Información Estadística y Geográfica de Jalisco,
Mexico

{rmaciel, vmlarios}@cucea.udg.mx,
{raul.beltran, jose.orizaga}@academicos.udg.mx
alejandro.bmartinez@alumnos.udg.mx,
edgar.cossio@iieg.gob.mx

Abstract. Faced with the imminent high fatal cyclist-car accident rate in the Guadalajara Metropolitan Area in recent years, it is necessary to implement mechanisms to improve the safety of cyclist mobility. This research analyzes the principal factors and patterns that cause cyclist-car accidents at intersections in the Guadalajara Metropolitan Area through machine learning algorithms and statistical methods. The data show that the most dangerous intersection consists of one main street and a street. As well the type of vehicle most involved in accidents with cyclists is public transport. The analysis shows that factors such as the speed limit and the lack of traffic lights increase the risk on some roads. It was also found that public transport is hazardous in Street-street type orthogonal intersections. Also, private vehicles are the leading cause of accidents in non-orthogonal intersections consists of one main street and a street where one road involved has three or more lanes.

Keywords. Fatal cyclist-car accidents, intersections, sustainable mobility, statistical analysis, artificial intelligence, pattern recognition, smart city.

1 Introduction

One of the premises of a smart city is generating clean and sustainable mobility where the citizens can move in an agile and safe way. Cyclist mobility can be part of one

solution in clean mobility improvement. In the Guadalajara Metropolitan Area (GMA), the governmental and academic authorities are working to transform mobility into smart and sustainable mobility. The city of Guadalajara has more than 100km of bike paths [1], the same ones that are growing. Also, the government, academia, and industry are creating mechanisms to promote bicycle mobility; one of them is the IoP (Internet of People) Jalisco [2]. According to the World Health Organization, bikers are part of the most vulnerable sector of the public via [3].

Therefore, bikers have a high possibility to die or suffer serious harm to their health in a traffic accident. There are three different scenarios for cyclist accidents. A single-bicycle accident, when a cyclist falls or crashes with an object [4]. Bicycle-bicycle accidents [5], and cyclist-car accidents [6].

This work focuses on cyclist-car accidents because those are the most reported and dangerous incidents for the cyclist's community [6, 7]. Despite the continuous growth of cycling infrastructure in the GMA, the biker community has many fatal accidents year by year. The average of lost lives was 23 per year from 2009 to 2019 [8].

Furthermore, when it seemed that the accident rate tended to decrease, the year 2019 compared with 2018 presented an increase of 53.8% of deaths [9] (see Fig. 1). This paper reviews the cyclist-car accident literature, emphasizing the intersections of streets and entrances and exits of car parks and shops.

Since, from different cities globally, the crossroads and intersections are the most frequent places for cyclist-car accidents [6] [7] [10], the main driver of this work is to find factors and patterns with a high impact on GMA fatal cyclist-car accidents at intersections.

We propose, based on historical data, identifying possible factors and accident patterns with three goals. First, to help the government decide better how to allocate their resources to improve the overall safety for citizens moving on bicycles. Second, provide simple but valuable information to citizens to increase their awareness of where patterns of accidents are part of their trip. Third, the possibility of deciding where to place internet of things objects in dangerous crossroads to mitigate as accidents as possible.

This work is organized in the following sections: section two discusses the literature found on bicycle accidents in cities, and the different approaches, datasets used, and results.

Section three relates the GMA problem, explains the primary data to use, and proposes an analysis with contingency tables and correspondence and clustering machine learning algorithms to identify the most common incidents. Section four discusses the approach and found results. Finally, section five concludes and presents the following steps on this work.

2 Literature Review

In a study that the UDV (German Insurers Accident Research) realized, 407 cyclist-car accidents were analyzed [6]. The work shows that when the car travels straight ahead or turns left or right, and the bicycle is coming from the right or left, the car's average speed is 19 km/h to 23 km/h. When the bicycle moves in the same direction before the impact, the average speed is 51 km/h. It can seem that accidents in which the cyclist is

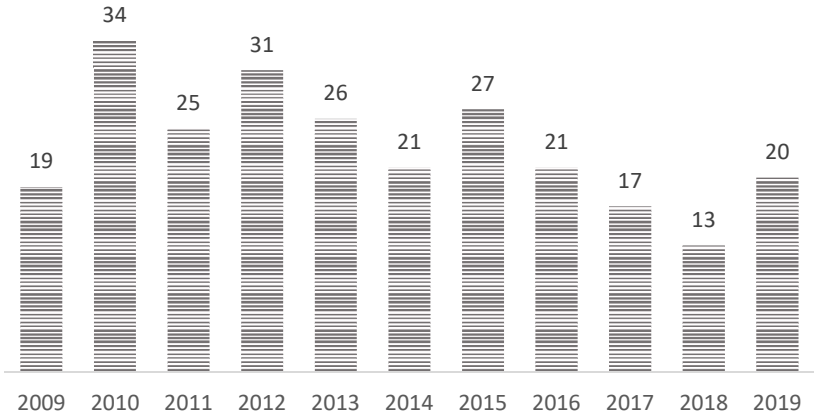


Fig. 1. Death of cyclists on public roads from the year 2009 to 2019 [8].

coming from the right or left usually occur at the entrances or exits of properties or parking lots and intersections of streets.

The accidents were caused for different reasons like vision obstacles or breaking the rules of the road. In Denmark, different bicycle facility layouts in signalized intersections were studied to compare their safety based on 80 hours of video recording for each intersection [11]. The results show that the collision risk is proportional to the traffic intensity. The risk situations occur in the most significant proportion when the driver turns to his right.

The safer layouts increase the distance between the road and the bike path, and car drivers must brake before turning. That situation allows drivers to see the cyclists with more time and a better angle vision. A study analyzed 3350 cyclist-car crashes from New York City to determine the relationship between intersection angle, street width, and accident severity [10]. Approximately 60% of crashes occurred in an intersection, and the rest in a non-intersection part of the street. The intersections with angle orthogonal ($85<\times<95$ grades) are safer than no orthogonal intersections ($0<\times\leq 85$ or $95\leq\times<175$ grades).

In a no orthogonal intersection, the visual angle is lower, and drivers and cyclists have less time to react and evicted a collision. The streets that have bicycle facilities had more than 50% fewer accidents. Also, the analysis showed that the accidents where a heavy truck or a public transport unit are involved result in a more severe injury compare with small vehicles. Street width was not significant in the severity, but the accidents at night have a significantly higher risk. In the Thomas Richter and Janina Sachs research [12], 873 cyclist-car accidents with turning vehicles and cyclists driving straight ahead were studied.

A third of the accidents correspond to vehicles that turn left, while two parts with a right turn. Three out of every 4 of these accidents occur at intersections with traffic lights. Also, one in 10 turning accidents between vehicles and cyclists happened with the participation of trucks. In the same study with video-recording monitoring, the

Table 1. Structure of database records [8].

variable	class
Sex	M
	F
Type of vehicle	Public transport
	Private car
	Truck
	Unidentified
Age	0-19
	20-39
	40-59
	60+
	Unidentified
Number of lanes per direction	1c
	2c
	3c+
Speed limit	30km/h
	50km/h
	60km/h
	80km/H
Number of directions	1s
	2s
Number of entrances and exits	3e
	4e
	5e+
Cycling infrastructure	Yes
	No
Public transport routes	Yes
	No
Orthogonal intersection	Yes
	No
Roundabout	Yes
	No
Traffic light intersection	Yes
	No
Type of intersection	Main street-main street
	Main street-street
	Street-street

situation that creates most conflicts is when a vehicle starts to drive after the red light, and a cyclist passes through the intersection without stopping. A German Federal Highway Research Institute study analyzed 120 accidents between right-turning trucks and straight driving cyclists [13].

Accidents at crossings and intersections between right-turning trucks and cyclists that move straight are particularly severe if the cyclist is hit and, consequently, overrun. Even though this type of accident is not frequent at intersections, 1 in 10 accidents represents a death.

In this kind of accident, bicycles travel at less than 20 km/h in 80% of accidents, and the trucks travel at less than 30 km/h in 90% of the cases. Bicycles and trucks do not change their speed at the time of the crash. Another study [14] indicates that this type

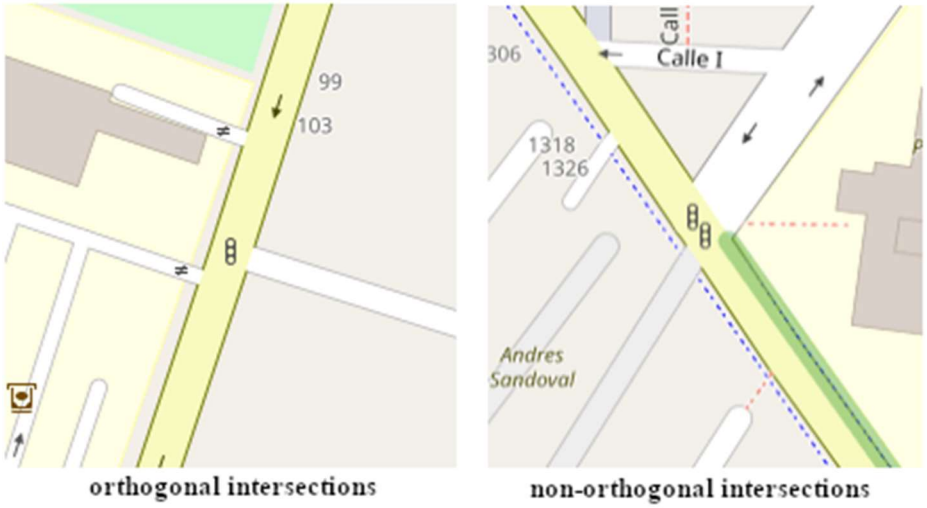


Fig. 2. The intersection is orthogonal when the street angle is between 85 and 95 degrees, and otherwise, the intersection is non-orthogonal. The left side of the figure shows orthogonal intersections, while the right side contains non-orthogonal intersections [9].

of accident occurs mainly at signposted intersections and roundabouts in urban areas. Within urban areas, roundabouts are a type of intersections of particular interest.

For example, a ten-year analysis of reported crashes from 2001 to 2011 in New Zealand showed that cyclists are involved in 28% of roundabout crashes [15]. Furthermore, in an analysis of cyclist risk carried out in Great Britain (2657 individuals were injured at an intersection), roundabouts are the type of intercession with the highest risk index [16]. The factors with the most significant impact on risk in roundabouts were vehicle speed and the streets' width and importance. Also, in a study carried out in the cities of Vancouver and Toronto, where 211 crashes between motorized vehicles and bicycles were analyzed, it confirmed that roundabouts are the type of intersection with the highest risk [17].

In this work, the relevant variables were: the speed of motorized vehicles (more than 30 km/h the risk increases), traffic flow (the more significant the flow of traffic of both vehicles and cyclists increases the risk), the diameter of roundabouts (the smaller the diameter, the greater the risk), the cycling infrastructure (obstacles that separate cyclists from vehicles reduce the chance) and the downward slope of the street on which the driver reaches the intersection.

Finally, an analysis carried out on accidents from 2004 to 2016 about roundabouts in Russia found that the number of exits from a roundabout is highly relevant to the probability that a cyclist suffers an accident with a motorized vehicle [18]. That situation is explained by a higher concentration of conflict points and a relationship with an unfavorable geometry, especially in small roundabouts.

The most common high-risk behavior among bicyclists and motor vehicle drivers at intersections is to ignore red lights or stop signs.

Indeed, cyclists tend to invade other public road areas such as the pedestrian zone to prioritize crossing and thus avoid the red light [19].

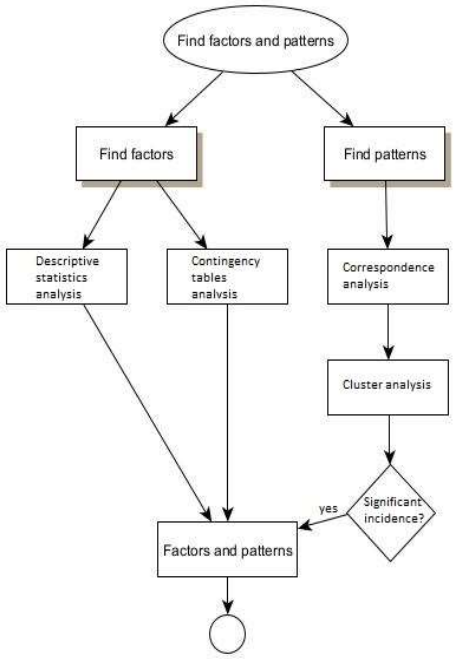


Fig. 2. Diagram of the methodology.

3 Fatal Cyclist-Car Accidents at Intersections Analysis from the Guadalajara Metropolitan Area

From January 2009 to Jun 2021, the White Bicycle organization registered 274 deaths of cyclists in the Guadalajara Metropolitan Area [8]. Two hundred thirty-nine records were identified as a cyclist-car accident, six as falls, two collisions with objects, and 27 where the cause is unknown.

Of the 239 fatal cyclist-car accidents, 56 occurred on a stretch without intersections, five times the location was not identified, and 179 at an intersection. For the analysis, 179 cyclist-car accidents at intersections were used.

The Fatal cyclist-car accidents database initially contains the variables of Sex, Type of road, Age, Type of vehicle, and Location of the accident site.

Then, to obtain the most significant number of variables found in the literature review, the street view of Google maps and the map of the Moovit platform were used for each location. With these tools, the variables Number of lanes per direction, Speed limit and Number of directions of the most significant road involved, and the Number of entrances and exits in each intersection were added.

In addition, the binary variables of the Presence of cycling infrastructure, Presence of public transport routes, Orthogonal intersection (Fig. 2 explains orthogonal intercessions), Roundabout, and Traffic light intersection were also added.

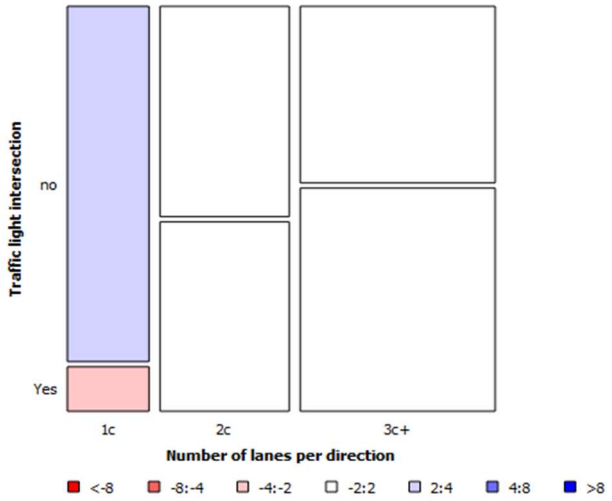


Fig. 3. The blue mosaic shows that there were more accidents than expected between classes no and 1c, while the pink mosaic shows that fewer accidents than expected were found between the same classes. The relationship of these three classes can show that traffic lights influence the number of accidents on one-lane streets per direction.

Finally, the variable Type of road that describes only one road involved in the intersection was modified to express two roads and was called "Type of intersection." Table 1 shows the structure of the database records.

3.1 Methodology

In order to find factors and patterns, the data analysis includes four different techniques. First, a descriptive statistical analysis highlights the main characteristics of fatal accidents at intersections from the GMA. Then a contingency table with Pearson's standardized residual calculation analysis is performed to find relations between the different classes of each combination of two variables. Descriptive statistics analysis and the relationships found in the contingency tables can reveal essential factors of the GMA accidents between cyclists and motorized vehicles.

In the contingency tables, if a class is present in a greater or lesser quantity than expected, it may be a triggering factor in accidents. For example, suppose in the contingency table between the variables Speed limit and Number of lanes per direction, there are more accidents than expected in the 80 k/m class. In that case, this may mean that the speed limit is a factor that affects the occurrence of fatal accidents with cyclists. In the end, correspondence analysis and a clustering algorithm are executed to identify associations between factors.

With correspondence analysis, it is possible to identify strong associations between classes. Subsequently, the presence of said associations in the centroids of the clusters is sought to formulate possible patterns.

To recognize an association of factors as a pattern, this association must show the presence of a phenomenon. Once defining an association of different classes as a

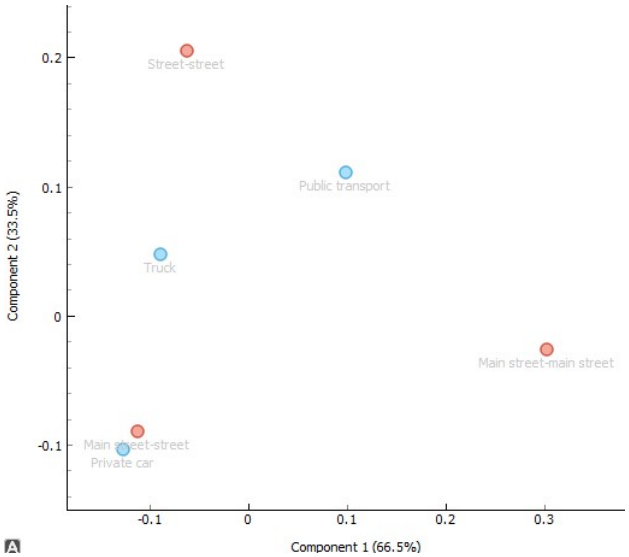


Fig. 4. Correspondence analysis between the Type of vehicle and Type of intersection variables. The red points represent the classes of the variable Type of intersection, and the blue points the classes of the variable Type of vehicle. In the figure, it is possible to see the closeness between the class's Private vehicle and the Main street-street type intersection.

Table 2. Centroids of the variables that present variation in the simple K-means of 2 clusters.

Attribute	Cluster 0	Cluster 1
Number of records	119	60
Type of vehicle	Private car	Public transport
Number of lanes per direction	3c+	1c
Orthogonal intersection	No	Yes
Traffic light intersection	Yes	No
Type of intersection	Main street- street	Street-street

pattern, it is corroborated that the pattern found has at least a 20% higher incidence than any other combination of classes of the variables involved. For example, suppose it is found that Public transport is mainly dangerous in intersections type Main street-street without-traffic light.

In that case, it must be verified that the number of accidents of this combination is at least 20% greater than that of a Private car or a Truck in the Main street-street without-traffic lights type of intersection. The above is in order to find patterns with significant incidence. Fig. 2 shows the process of the methodology.

3.2 Descriptive Statistics Analysis

The descriptive statistics analysis observed that 163 (91.06%) victims are male, and 16 (8.94%) are female [8]. In contrast to what was mentioned in [15, 16, 17], roundabouts

Table 3. Centroids of the variables that present variation in the simple K-means of 3 clusters.

Attribute	Cluster 0	Cluster 1	Cluster 2
Number of records	86	34	59
Type of vehicle	Public transport	Public transport	Private car
Number of lanes per direction	3c+	1c	3c+
Orthogonal intersection	Yes	Yes	No
Traffic light intersection	Yes	No	No
Entrances and exits	4e	4e	3e
Type of intersection	Main street-street	Street-street	Main street- street

are not the most dangerous type of intersection speaking about fatal cyclist-car accidents for the GMA. Because of the 179 incidents, only 7, equivalent to 3.91%, occurred in a roundabout. Furthermore, it was found that, in contrast to [18], increasing the number of entrances and exits from an intersection does not increase the risk of a fatal cyclist-car accident.

The predominant number of entrances and exits at intersections was 4 with an occurrence of 56.42%, followed by 3 with 39.66%, intersections with five or more entrances and exits had an event of 3.91%. On the other hand, it cannot assure that the existence of cycle lanes reduces the number of accidents, as mentioned in [12]. However, the records show that 91.06% of fatal accidents occurred at intersections without bicycle infrastructure.

As mentioned in [10], fatal accidents involving public transport and trucks are more frequent than accidents with small vehicles for the GMA. Accidents with public transport and trucks represent 56.43%.

Thus, public transport is the enemy number one for cyclists with 42.46% of incidents, followed by private cars with 40.78%. Similar to that mentioned in [16], the importance of the streets is of great impact since only 25.14% of the recorded intersections do not contain main roads.

The most common intersection consists of one main street and a street, present in 50.84% of the occasions. In addition, 99% of the intersections with at least one main street have public transport routes, and 98% are two-way traffic. Also, the number of lanes per street increases according to the importance of the streets in the intersection. It is detected that cyclists between the ages of 20 and 39 are most vulnerable to fatal accidents.

3.3 Contingency Tables Analysis

Contingency tables with Pearson's standardized residual calculation were used to identify the events that occur to a greater or lesser extent than expected. A relationship between two classes can show a causal factor in the increase or decrease of accidents, while a relationship of more than two classes of two variables can reflect the existence of a causal phenomenon.

The following deviations were found in the contingency table for each combination of two variables Using Orange software [20]. With a standardized Pearson residual of 3.3, 11.9 more accidents than expected were found in intersections where one street has a speed limit of 80 k/h and three or more lanes per direction. With a Pearson

standardized residual of -3.0, 11.9 fewer accidents than expected were found at intersections with traffic lights where the streets have one lane per direction.

Otherwise, with a standardized Pearson residual of 2.7, 12.1 more accidents than expected were found in intersections without traffic lights where the streets have one lane per direction (see figure 3). With a standardized Pearson residual of 8.0, 23.9 more accidents were found than expected in the type of intersection Street-street of one lane per direction.

With a standardized Pearson residual of 3.2, 6.5 more accidents than expected were found in the type of intersection Main street-main street, where the arteries' speed limit is 80 k/m. With a standardized Pearson residual of 6.3, 15.7 more accidents than expected were found in the type of intersection Street-street where the streets are one way.

3.4 Correspondence Analysis

For principal component analysis, the unsupervised visual correspondence analysis of Orange software was used. Correspondence analysis shows the existence of a relationship between classes of two or more variables. The relationship is represented in the form of visual proximity with coordinates of the cartesian plane.

As a result, the following most significant proximity between classes is observed: Private car of the Type of vehicle variable and 3e of the Number of entrances and exits variable; Public transport of the Type of vehicle variable and 4e of the Number of entrances and exits variable; Private car of the variable Type of vehicle and Main street-street of the variable Type of intersection (see in figure 4); 2c of the variable Number of lanes and Yes of the variable Presence of public transport routes; 1c and 2c of the variable Number of lanes with Yes of the variable Orthogonal intersection; 3c+ of the variable Number of lanes with No of the variable Orthogonal intersection; Yes of the variable Presence of public transport routes with 2s of the variable Number of directions; Yes of the variable Orthogonal intersection with 4e of the Number of entrances and exits variable; No of the variable Orthogonal intersection with 3e of the Number of entrances and exits variable.

3.5 Cluster Analysis

Finally, a couple of grouping was made with the simple k-means unsupervised algorithm of the Weka software. It was chosen to work with 2 and 3 clusters since most of the variables involved are nominal categorical of two or three classes. With this, it was possible to observe how the factors of each variable are distributed and associated in the centroids of the clusters. Statistical results of the distribution and centroids of the variables with variation are shown in Tables 2 and 3.

4 Results and Discussion

Based on the descriptive statistics analysis, the analysis with contingency tables, the correlational analysis, and the cluster analysis, the following patterns and principal factors could be found:

1. The closeness in the correspondence analysis between the classes Private car and the Main street-street, three or more lanes with non-orthogonal and the presence of these four classes as centroids in cluster 0 of the grouping in two clusters and cluster 2 of the grouping in 3 clusters indicate that: in the type of intersection Main street-street non-orthogonal where a road has three or more lanes per direction, the private car is the main one causing fatal accidents with cyclists.
2. In the grouping in 2 and 3 clusters, the classes' Public transport, Orthogonal intersection, type of intersection Street-street, and one lane per direction are centroids of a cluster for each grouping. In addition, in the correspondence analysis, the class two lanes and the presence of public transport show closeness, which indicates that: Public transportation is the most dangerous vehicle for cyclists who circulate at orthogonal Street-street intersections with one and two lanes per direction.
3. By finding more accidents than expected in intersections with one-lane streets without traffic lights and fewer accidents than expected in intersections with one-lane streets with traffic lights, and the presence of classes one lane and intersection without traffic lights as centroids in clusters 1 of the grouping in two clusters and cluster 1 in the grouping in 3 clusters indicates that: The presence of traffic lights at intersections where the streets have one lane per direction reduces the risk of a fatal accident when riding a bicycle.
4. More accidents than expected were found in Main street-main street intersections and roads with three or more lanes with a speed limit of 80 km/h.. Thus, the above tells us that intersections type Main street-main street and the intersections where at least one road has three o more lanes are more dangerous at a speed limit of 80 k/h.
5. The high incidence of 3 and 4 entrances and exits could explain drivers do not reduce their speed at intersections with few points of conflict, an event that, on the contrary, occur in roundabouts. Therefore, although roundabouts have a high number of conflict points, they have little presence of fatal cyclist-car accidents.

5 Concluding Remarks and Future Work

The descriptive statistics analysis shows that the type of intersection with the highest risk of a fatal accident is formed by one main street and a street. Public transport is the kind of vehicle most involved in fatal accidents with cyclists.

Also, the following patterns could be found: in the intersection main street-street non-orthogonal where a road has three or more lanes per direction the private vehicle is the main one causing fatal accidents with cyclists; public transportation shows to be a vehicle hazardous for cyclists who circulate at street-street orthogonal intersections with one and two lanes per direction; the presence of traffic lights at intersections where the streets have one lane per direction reduces the risk of a fatal accident when riding a bicycle; roads with a higher speed limit increase the risk of a cyclist-car fatal accident at intersection type main street-main street and the intersections where at least one road has three o more lanes.

In future work, the analysis will be extended to include variables such as the traffic index and the flow of cyclists and include new techniques such as the market basket

analysis. In addition, it will seek to classify the intersections of a polygon in the GMA based on the risk of suffering a fatal cyclist-car accident.

References

1. Vega, I. P.: Más de 100 km de ciclovías en Guadalajara, motivo para "celebrar" en el Día Mundial de la Bicicleta. UDG TV (2020)
2. Hoozie—Users. (s. f.). Hoozie. Recuperado 8 de octubre de 2021, de https://ww w.hoozie.io/?lang=en
3. WHO. (s. f.). Global status report on road safety 2018. Recuperado 7 de octubre de 2021, de https://www.who.int/publications-detail-redirect/9789241565684
4. Schepers, P., Klein Wolt, K.: Single-bicycle crash types and characteristics. Cycling Research International, vol. 2, pp. 119–135 (2012)
5. Wallentin, G., Loidl, M.: Bicycle-bicycle accidents emerge from encounters: An agent-based approach. Safety, vol. 2, no. 2, pp. 14 (2016)
6. Kuehn, M., Hummel, T., Lang, A.: Cyclist-car accidents–their consequences for cyclists and typical accident scenarios.
7. Glász, A., Juhász, J.: Car-pedestrian and car-cyclist accidents in Hungary. Transportation Research Procedia, vol. 24, pp. 474–481 (2017) doi: 10.1016/j.tr pro.2017.05.085
8. Datos Bici Blanca. (s. f.). Datos abiertos. Recuperado 7 de octubre de 2021, de https://docs.google.com/spreadsheets/d/1fXJGPruV8Flzvb_fKoQ39YmqASXXfva8hfgyI8 0arR0/edit?usp=embed_facebook
9. Carapia, F. (s. f.). Matan a más ciclistas pese a las ciclovías. Recuperado 7 de octubre de 2021, de https://www.reforma.com/matan-a-mas-ciclistas-pese-a-las-ciclovias/ar1861611
10. Asgarzadeh, M., Verma, S., Mekary, R. A., Courtney, T. K., Christiani, D. C.: The role of intersection and street design on severity of bicycle-motor vehicle crashes. Injury Prevention, vol. 23, no. 3, pp. 179–185 (2017) doi: 10.1136/injuryprev-2016-042045
11. Madsen, T. K., Lahrmann, H.: Comparison of five bicycle facility designs in signalized intersections using traffic conflict studies. Transportation Research Part F: Traffic Psychology and Behaviour, vol. 46, pp. 438–450 (2017) doi: 10.10 16/j.trf.2016.05.008
12. Richter, T., Sachs, J.: Turning accidents between cars and trucks and cyclists driving straight ahead. Transportation Research Procedia, vol. 25, pp. 1946–1954 (2017) doi: 10.1016/j.trpro.2017.05.219
13. Seiniger, P., Gail, J., Schreck, B.: Development of a test procedure for driver assist systems addressing accidents between right turning trucks and straight driving cyclists (2015) doi: 10.13140/RG.2.2.36586.72649
14. Pokorny, P., Drescher, J., Pitera, K., Jonsson, T.: Accidents between freight vehicles and bicycles, with a focus on urban areas. Transportation Research Procedia, vol. 25, pp. 999–1007 (2017) doi: 10.1016/j.trpro.2017.05.474
15. Tan, T., Haque, S., Lee-Archer, L., Mason, T., Parthiban, J., Beer, T.: Bicycle-friendly roundabouts: A case-study. Journal of the Australasian College of Road Safety, vol. 30, no.4, pp. 67–70 (2019) doi: 10.3316/informit.032179159989363
16. Aldred, R., Kapousizis, G., Goodman, A.: Association of infrastructure and route environment factors with cycling injury risk at intersection and non-intersection locations: A case-crossover study of Britain. International Journal of Environmental Research and Public Health, vol. 18, no. 6 (2021). doi: 10.3390/ij erph18063060
17. Harris, M. A., Reynolds, C. C. O., Winters, M., Cripton, P. A., Shen, H., Chipman, M. L., Cusimano, M. D., Babul, S., Brubacher, J. R., Friedman, S. M., Hunte, G., Monro, M., Vernich, L., Teschke, K.: Comparing the effects of infrastructure on bicycling injury at intersections and non-intersections using a case–crossover design. Injury Prevention, vol. 19, no. 5, pp. 303–310 (2013). doi: 10.1136/injury prev-2012-040561

18. Hollenstein, D., Hess, M., Jordan, D., Bleisch, S.: Investigating roundabout properties and bicycle accident occurrence at swiss roundabouts: A logistic regression approach. ISPRS International Journal of Geo-Information, vol. 8, no. 2 (2019) doi: 10.3390/ijgi8020095
19. Kummeneje, A. M., Rundmo, T.: Attitudes, risk perception and risk-taking behaviour among regular cyclists in Norway. Transportation Research Part F: Traffic Psychology and Behaviour, vol. 69, pp. 135–150 (2020) doi: 10.1016/j.trf.2020.01.007
20. Demšar, J., Curk, T., Erjavec, A., Gorup Č., Hočevar, T., Milutinovič, M., Možina, M., Polajnar, M., Toplak, M., Starič, A., Stajdohar, M., Umek, L. L., Jure, Z., Marinka, Z., Zupan, Z. B.: Orange: Data mining toolbox in Python. Journal of Machine Learning Research, vol. 14, no. 35, pp. 2349–2353 (2013)

ISSN 1870-4069

Artificial Intelligence and Component-Based Software Engineering: A Systematic Mapping Study

Bruno A. López-Luján, Ángel J. Sánchez-García, Karen Cortés-Verdín

Universidad Veracruzana,
Facultad de Estadística e Informática,
Mexico

{angesanchez, kcortes}@uv.mx, bruno1529@live.com.mx

Abstract. Industry 4.0 has changed the way businesses and organizations operate. Faced with this change, organizations must identify the technologies that best meet their needs in order to invest in them, especially in the way of developing software products that meet the needs that this revolution involves. Component-Based Software Engineering (CBSE) seeks reusing to define, implement, and compose weakly coupled software components in systems. This research proposes to carry out a Systematic Mapping Study (SMS), based on the Kitchenham and Charters guide, to associate Artificial Intelligence (AI) approaches to CBSE problems. In this SMS, 38 primary studies were selected. From these studies, it was observed that Artificial Intelligence has supported to CBSE through 16 algorithms and 17 AI techniques, highlighting optimization techniques such as Evolutionary Algorithms. On the other hand, 28 problems were found within 14 CBSE activities where the activity most addressed by AI has been Software Reliability.

Keywords: Artificial intelligence, component-based software engineering, systematic mapping study, optimization.

1 Introduction

Industry 4.0 is identified by the emergence of new technologies such as Robotics, Analytics, Artificial Intelligence (AI), Nanotechnology and the Internet of Things (IoT), among others. However, all these areas converge in the need for software development that computes and manages the data that is generated every day.

Recently, Software development has seen in Component-Based Software Engineering (CBSE) an opportunity to reduce software development times and therefore release and delivery times. CBSE is “an approach that proposes the development and evolution of a software system through the selection and composition of components" [1].

In addition, CBSE is known to be a specialized way of creating software from existing components, allowing the reuse of previously used components, where these components are selected for use. On the other hand, Artificial Intelligence (AI) is a very

broad field that began shortly after the Second World War. Throughout history, AI has followed four approaches: systems that think like humans, systems that think rationally, systems that act like humans, and systems that act rationally [2].

The Software Engineering and Artificial Intelligence disciplines have collaborated with each other in various areas, such as Requirements Classification [3], Requirements Prioritization [4], Component-Based Software Engineering [5], among others. There are Artificial Intelligence techniques that are used to solve some of the problems currently present in the different activities of Component-Based Software Engineering.

The applications of AI techniques in CBSE have shown successful results [6, 7, 8] or at least, raise proposals for future research [9, 10]. However, the literature indicates that Component-Based Software Engineering has problems that, although there are proposals to solve them, have not been fully resolved. Performing a manual search, no research papers have been found that succinctly cover all the applications of AI techniques for solving the problems that exist within Component-Based Software Engineering.

This is important since researchers currently have difficulty obtaining information from the AI techniques used in CBSE, so this work will provide a reliable source of information for them. The Systematic Mapping Study (SMS) will allow knowing the current state of Artificial Intelligence applications in Component-Based Software Engineering. In addition, it will benefit researchers, to strengthen the domain of the subject with the applications, the result of said applications, problems addressed, and proposals and solutions found.

This in order to facilitate the creation of research works in this area and the development of innovative contributions. This paper is organized as follows: Section 2 presents the background and related work. In Section 3, the method followed to carry out this SMS is described. Section 4 presents the final results. Finally, Section 5 draws conclusions and defines future work.

2 Background and Related Work

In the manual search of related works, and according to the authors' knowledge, only a paper was found as a systematic mapping study related to the subject. In 2018, Diwaker et al. [11] carried out a research where they made a list of all the applications of techniques and algorithms of the Soft Computing category belonging to Artificial Intelligence on different problems (mainly the prediction of software reliability) found within the Software Reliability activity pertaining to Component-Based Software Engineering.

The list of the soft computing techniques found were “Genetic Algorithm (GA), Neural-Network (NN), Fuzzy Logic, Support Vector Machine (SVM), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Artificial Bee Colony (ABC)” [11] (only soft computing).

The research covered the needs, the area of opportunity that exists, as well as the difficulties and problems it presents, and the problems that have already been addressed in Software Reliability. This text describes the techniques and algorithms that have been used in Software Reliability, approached from Soft Computing. Finally, the importance of activities within the CBSE was exposed.

Table 1. Keywords.

Concept	Keywords
Component-Based Software Engineering	Component-Based Software Engineering, Component Based Software Engineering, Component-Based Software, Component Based Software
Artificial Intelligence	Artificial Intelligence, methodology, algorithm, technique

Table 1. Inclusion criteria.

IC	Description
IC1	The paper published from 2011 to May 2021 will be included
IC2	The paper that is written in English will be included.
IC3	The paper that in the title or abstract includes at least two key search concepts will be included.
IC4	The article that after reading the abstract gives the impression that it solves the research questions will be included.

3 Research Method

To carry out this research work, the guidelines for Systematic Literature Reviews (SLR) in Software Engineering, which were proposed by Kitchenham & Charters [12], were used as a guide. Although this guide is focused on SLR, it was decided to carry out a Systematic Mapping Study (SMS) to relate Artificial Intelligence techniques with CBSE problems as shown in the research questions. The method is presented in two stages: planning and execution.

3.1 Planning

Research Questions. The research questions (RQs) relating to this SMS are:

- **RQ1:** What are the Artificial Intelligence techniques that have been reported in the research of the CBSE area?
- **RQ2:** What have been the results in the applications of these techniques?

The motivation of RQ1 is to present the current state of AI algorithms and techniques applications at the CBSE. On the part of the RQ2, the motivation is to present the results of the applications of AI techniques and algorithms on the activities and problems of the CBSE to identify areas of opportunity.

Search strategy. An automated search was performed, searching for the primary studies among the resulting papers that demonstrated the selected information sources. All of this was done from a defined search string.

Keywords. Once the research questions were defined, the keywords were defined as shown in Table 1. The words "Artificial intelligence" and "Component-based software

Table 3. Exclusion criteria.

EC	Description
EC1	Papers that are not available will be excluded.
EC2	Papers that are only available in the form of slides, posters, books and technical reports will be excluded.
EC3	Papers with duplicate research will be excluded.

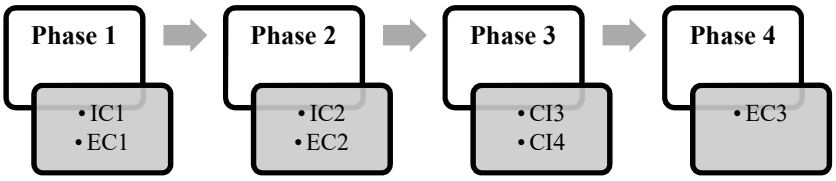


Fig. 1. Primary studies selection process.

engineering" were located as the most important terms for this research. Similar or related terms were added for each search keyword. The terms AI and CBSE were not included since not all authors use these abbreviations.

Search string. The keywords allowed the creation of the search string that was used in this investigation. After trying different chains of remaining as follows:

("Component-Based Software Engineering" OR "Component-Based Software" OR "Component Based Software" OR "Component Based Software Engineering") AND ("Artificial Intelligence" OR "Methodology" OR "Algorithm" OR "Technique")

Information sources. The information sources used for the search and selection of primary studies were four databases, being digital libraries: IEEE Xplore, ACM Digital Library, ScienceDirect and SpringerLink. These information sources were selected because they are repositories of articles from the computer science area and related disciplines.

Primary study selection criteria. Table 2 and Table 3 show the inclusion and exclusion criteria used to select the primary studies.

Primary study selection procedure. The phases that were applied to the resulting papers in the automated search for the selection of primary studies are found in Fig. 1.

Quality assessment. The quality evaluation was considered feasible to avoid threats to the quality of the information obtained as results of this research. For this point, the quality evaluation criteria were defined to have a quantifiable and objective result shown in Table 4.

These criteria are made up of eight questions applicable to each paper selected as the primary study, where 1 point will be given if it meets the criterion quality assessment or 0 if not compliant. The decision was made to select the papers that have a minimum of five points for use in this research. In this way we affirm that the articles used in this research are reliable and of quality.

Table 2. Quality evaluation criteria.

ID	Question
1	Are the objectives of the paper described?
2	Does the study mention papers or proposals already published?
3	In the related work section, does the study mention at least 3 papers that have been published up to five years before publication?
4	Does the study use public datasets?
5	Is the methodology used described in the study?
6	Was the research process adequately documented in the study?
7	Do the results prove the achievement of the objectives?
8	Are the conclusions of the study clear?

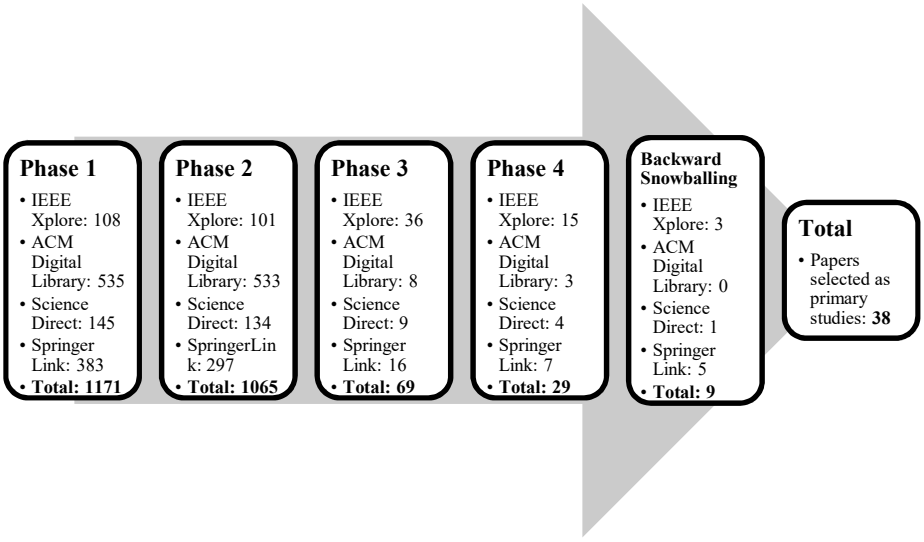


Fig. 2. Primary studies selection phases.

Data extraction. The template shown in Table 5 shows the data extracted from each primary study. Two types of data were extracted from each primary study and deemed necessary to answer the research questions.

The first seven data correspond to the details of the publication and the last two data are directly linked to RQ1 and RQ2, these data can answer both research questions.

3.2 Execution

The search string applied in the information sources yielded a total of 6,452 papers. As shown in Fig. 2, after applying all the phases of the selection process for primary studies, a result of 29 papers was obtained. Once the papers resulting from the selection procedure of primary studies shown in Fig. 1 were obtained, *Backward Snowballing* was applied to each of them, giving a result of 9 papers. Finally, a total of 38 papers

Table 5. Data extraction template.

Kind of data	Data extracted
Publication details	Title
	Authors
	Year
	Source
	Publication type
	Reference
	Abstract
Context	AI technique used
	CBSE activity and problematic addressed

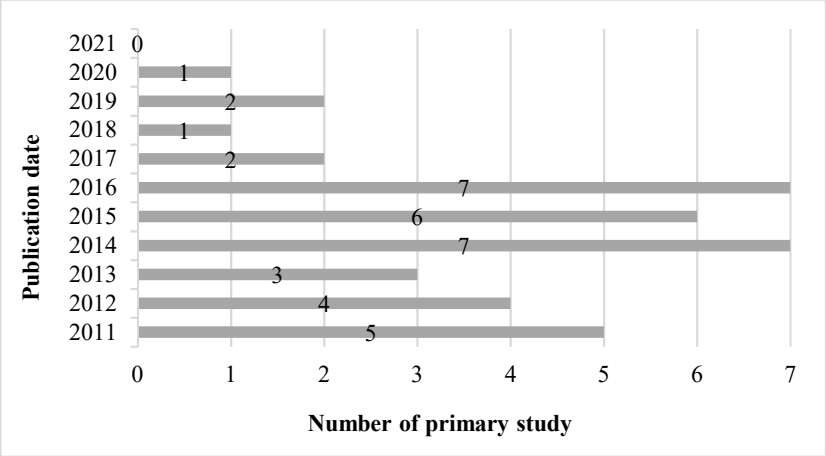


Fig. 3. Number of papers per year publication.

selected as primary studies were obtained. The detail of these 38 primary studies is presented in Appendix A[1].

4 Results

Of a total of 6,452 papers that emerged from the application of the search string in the four information sources, only 38 papers were selected as primary studies. In Fig. 3, a trend can be seen by researchers in the publication of papers where CBSE is addressed using AI techniques in the years 2014 to 2016.

Only in these three years were published 20 of the 38. To answer the research questions, a Thematic Synthesis Process was followed. Thirty-eight papers were selected as primary studies.

[1] Appendix A. 38 primary studies:
https://drive.google.com/file/d/1NqaIAs5cc1fhHPov3Zdx4M3fhxYjh6Sm/view?usp=sharing

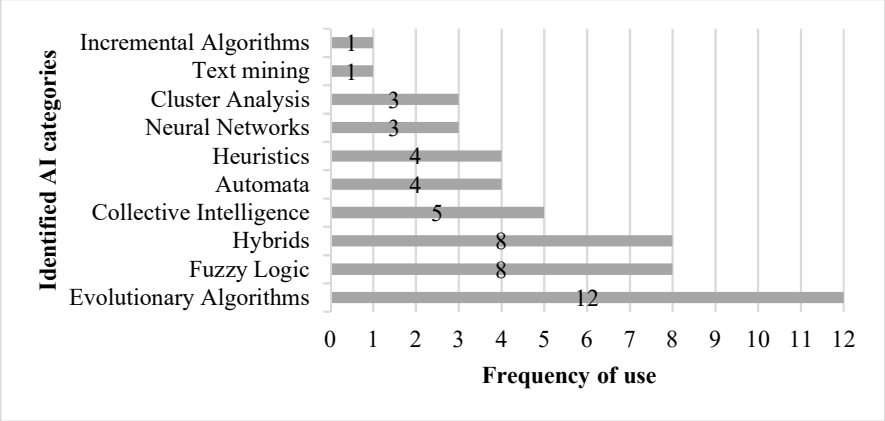


Fig. 4. Classification of AI algorithms and techniques.

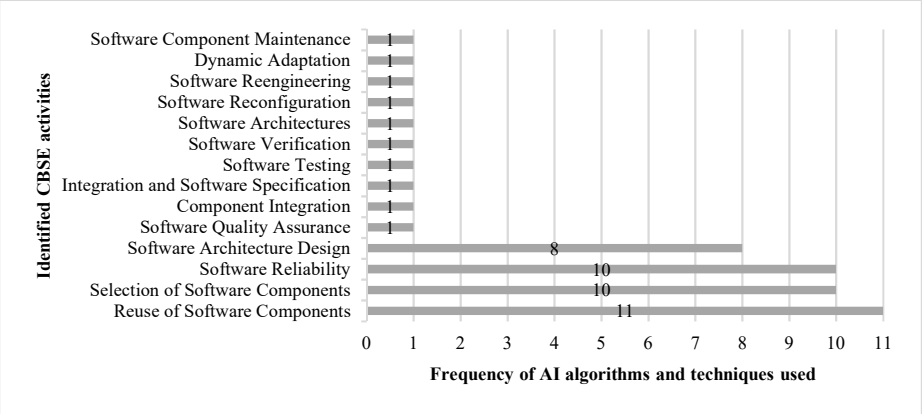


Fig. 5. CBSE activities and boarding frequency.

In these papers, 16 algorithms and 17 AI techniques were found to address different problems within the activities of the CBSE. The contributions of the AI on the CBSE were classified by categories.

To achieve this, techniques and algorithms that belonged to the same family were grouped as *Fuzzy Inference System* (FIS) and *Fuzzy Formal Concept Analysis* (FCA).

Both techniques were classified in *Fuzzy Logic*. To cite another example, *Evolutionary Algorithms* such as *Strength Pareto Evolutionary Algorithm* (SPEA2) and *S-Metric Selection Evolutionary Multiobjective Algorithm* (SMS-EMOA), as well as the rest of the evolutionary algorithms were classified in the same category. This classification can be seen in Fig. 4. Furthermore, the frequency of use of each of these categories on the CBSE can be seen. Regarding the problems and activities of the CBSE.

Within the primary studies 14 CBSE activities were found that were addressed using AI algorithms and techniques. Fig. 5 shows the activities found and the frequency of the AI approach. In Fig. 6, the problems within the identified activities are exposed. A trend is shown by researchers in the Reuse of Software Components approach, being

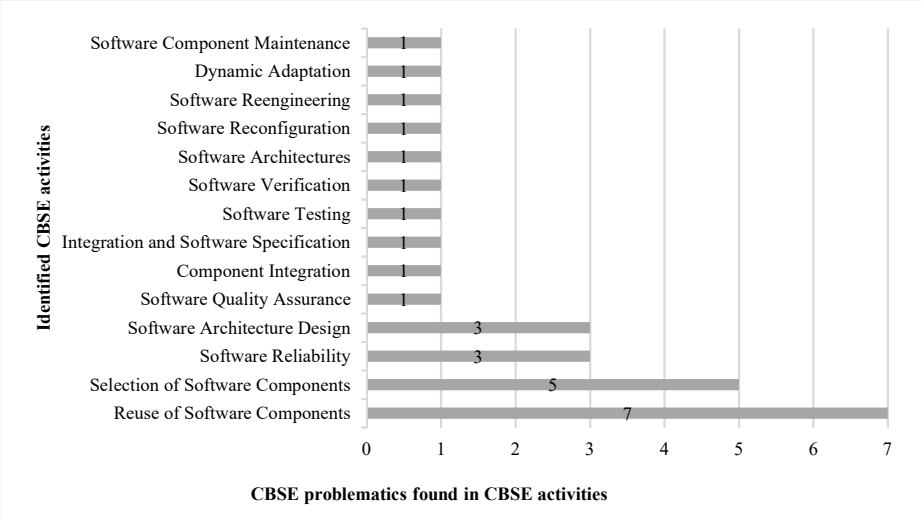


Fig. 6. Problematics found in CBSE activities.

the activity, most approached with proposals aimed at solving 7 problems. In response to RQ1, the use of 16 algorithms and 17 AI techniques on different issues within the activities of the CBSE was found.

We can expose the use of Genetic Algorithms (GA) which reported four approaches to solve different problems of the CBSE. Adaptive Neuro-Fuzzy Inference System (ANFIS), like GA reported four approximations. Finally, Fuzzy Logic (FL) with three proposals, together with GA and ANFIS add up to just over 33% of the proposals to solve problems or optimize CBSE activities.

This analysis demonstrates the variety of AI algorithms and techniques that have been reported in the current state of applications in the CBSE area, since the two techniques and the aforementioned algorithm were the most used by researchers in this research work.

As shown in Fig. 7. Evolutionary Algorithms (EA) were the most common algorithms for the approach to software reliability and the selection of the software component. However, in this category it is also proposed to address Software Testing and Software Architectures. In this way, Algorithms of evolution proves to be the category with flexibility in the contributions of AI on CBSE.

On the other hand, Text Mining and Incremental Algorithm only reported a contribution to address the activities of the CBSE, these being Integration and Specification of Software and Selection of Software Components respectively. Based on the results obtained from the Systematic Mapping Study and to respond to the RQ2, 38 studies are reported that address the CBSE through AI to address different activities and problems of this.

Each of the studies clearly describes its objectives and concisely explains the approach to a CBSE activity through a solution proposal, as well as other previously published studies to take them as a reference or compare them with their solution proposals. However, not all the selected primary studies mention updated articles

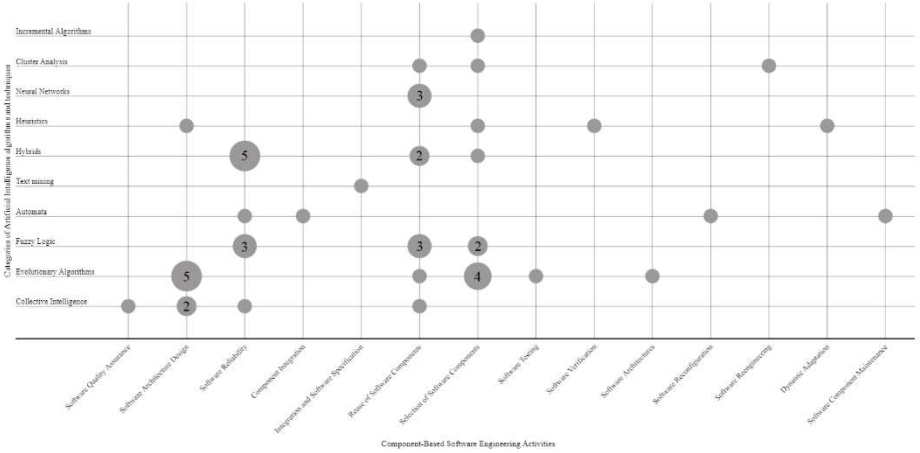


Fig. 7. Frequency of approach of AI on CBSE.

(taking as updated papers those papers published up to five years before the publication of the primary study) for comparison.

Selected primary studies use specific datasets, but not all datasets used are public or freely accessible. On the other hand, the selected primary studies describe the proposed methodology textually, graphically and concisely for a greater understanding of the reader. They also adequately address the proposal research process in each of them.

5 Conclusions and Future Work

A Systematic Mapping Study (SMS) was carried out to answer two research questions, applying the method proposed in [12] in four sources of information, where 38 articles selected as primary studies reported contributions on CBSE using AI techniques and algorithms. Once the results of this research work have been analyzed, classified and compared, it can be noted that there is a great tendency for researchers in CBSE to use Evolutionary Algorithms, Fuzzy Logic techniques, as well as algorithms and heuristic techniques to address problems pertaining to the different activities of Component-Based Software Engineering.

In addition, this research shows that the CBSE activities most addressed by Artificial Intelligence are: Component Reuse, Component Selection, and Software Reliability. The results obtained indicate that Artificial Intelligence addressed Component-Based Software Engineering 38 times through 16 algorithms and 17 AI techniques, solving 28 problems found within 14 CBSE activities.

In this work, it was shown that Artificial Intelligence and Software Engineering, specifically the area of Component-Based Software Engineering. AI supports several activities in the CBSE area, such as the choice of the best component, reliability estimation models, text mining, among others, which benefit in better reuse, less expensive and more accurate.

So, there is a lot of research to be done in this field in collaboration between areas, to ensure faster and more agile software development, as well as high quality, which is

fundamental in Industry 4.0. As future work, it is proposed to explore different optimization approaches to compare the results with those provided in the primary studies. This is because although they are addressed problems, they are not solved problems.

References.

1. Vale, T., Crnkovic, I., Santana de Almeida, E., da Mota Silveira Neto, P. A, Cavalcanti, Y. C., Romero de Lemos Meira, S.: Twenty-eight years of component-based software engineering. Journal of Systems and Software, vol. 111, pp. 128–148 (2016)
2. Russell, S. J., Norvig, P., Davis, E.: Artificial intelligence: A modern approach. Upper Saddle River, NJ: Prentice Hall (2003)
3. Hey, T., Keim, J., Koziolek, A., Tichy, W. F.: NoRBERT: Transfer learning for requirements classification. In: IEEE 28th International Requirements Engineering Conference (RE), pp. 169–179 (2020)
4. Kifetew, F., Munante, D., Perini, A., Susi, A., Siena, A., Busetta, P.: Dmgame: A gamified collaborative requirements prioritization tool. In: IEEE 25th International Requirements Engineering Conference (RE), pp. 468–469 (2017) doi: 10.1109/R E.2017.46
5. Xie, T.: The synergy of human and artificial intelligence in software engineering. In: 2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), pp. 4–6 (2013)
6. Su-Wei, G.: Software component retrieval method based on PSO-RBF neural network. In: 2nd International Conference on Computer Engineering and Technology, vol. 7, pp. V7–339 (2010)
7. Khatri, S. K., Kaur, G., Johri, P.: Multi-level selection of reusable software components. In: Proceeding of 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 67–71 (2016)
8. Khode, S. G., Bhatia, R.: Improving retrieval effectiveness using ant colony optimization. In: Proceeding of International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 737–741 (2009)
9. Cortellessa, V., Potena, P.: How can optimization models support the maintenance of component-based software? In: Proceeding of 1st International Symposium on Search Based Software Engineering, pp. 97–100 (2009)
10. Vodithala, S., Pabboju, S.: A dynamic approach for retrieval of software components using genetic algorithm. In: Proceeding of 6th IEEE International Conference on Software Engineering and Service Science (ICSESS), pp. 406–410 (2015)
11. Diwaker, C., Tomar, P., Poonia, R. C., Singh, V.: Prediction of software reliability using bio inspired soft computing techniques. Journal of medical systems, vol. 42, no. 5, pp. 1–16 (2018)
12. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Durham, UK: University of Durham (2007)

ISSN 1870-4069

Comparison of Machine Learning Algorithms for the Preventive Diagnosis of Robotic Arms for Palletizing

Julio Zambrano[1], Gerardo Reyes-Salgado[2]

[1] Instituto de Ingenieros de Morelos, Mexico

[2] Centro Nacional de Investigación y Desarrollo Tecnológico, Mexico

julio@jzambrano.xyz, gerardo.rs@cenidet.tecnm.mx

Abstract. Data from a robotic arm for palletizing of the Universal Robots (UR) brand was analyzed, which at the time of the experiment did not have an intelligent preventive maintenance system. Experiments were carried out with Machine Learning algorithms like Support Vector Machine (SVM), Gradient Boosting Machine (GBM) and Artificial Neural Network (ANN), as well as with ensembles of them. According to the evaluation metrics obtained, the ensemble of various algorithms turned out to be the tool for practical use that offers the best results towards the preventive diagnosis of robotic arms for palletizing.

Keywords: Machine learning, data analysis, robotic arms, preventive maintenance.

1 Introduction

The inclusion of systems with artificial intelligence for preventive maintenance in robotic systems, becomes a very useful tool for the prevention of economic losses, reduction of downtime of equipment and improvement of the quality of life of operators of these systems, by avoiding accidents due to failures that could have been prevented. An example of the above appears within the manufacturing industry in the United States, which in 2016 reported an expense of 50 billion dollars, just in maintenance and repairs [1].

Within this context are the robotic arms for palletizing of the brand Universal Robots (UR, Fig. 1.), which work cooperatively with operators of packaging and palletizing plants. Despite the fact that brands such as Omron, ABB and KUKA already have preventive maintenance systems implemented in their robotic arm systems, Universal Robots does not have this alternative yet, which creates an area of opportunity to look for solutions to this problem[1]. By continuously analyzing data from these robots while they are in operation, an intelligent preventive diagnostic system is able to identify

[1] Consulted on July 25, 2021 at https://www.universal-robots.com/

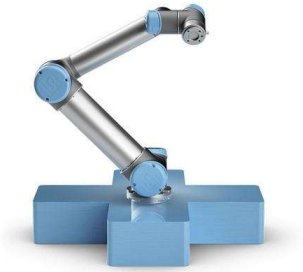


Fig. 1. Robotic arm of the brand Universal Robots.

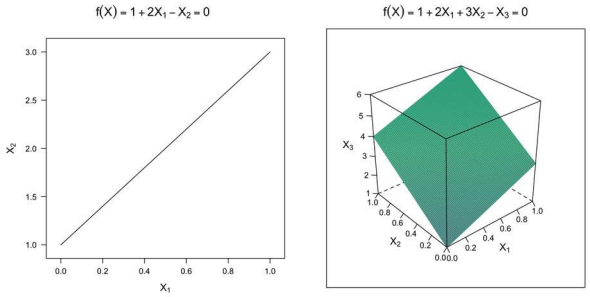


Fig. 2. Visual representation of hyperplanes in two and three dimensions using the SVM algorithm (Boehmke, 2020).

trends in the operation of these devices, with the help of Machine Learning algorithms and their ensembles. In addition to that, beyond this study for a particular opportunity in a specific brand, comes the interest to explore the use of Machine Learning algorithms on failure diagnostics, topic that can be translated to other fields of science and engineering. The most recent literature related to the practical use of Machine Learning algorithms in industrial preventive maintenance was studied in order to find a solution to this problem.

Based on the above, the algorithms of Support Vector Machines (SVM, Fig. 2.), Gradient Boosting Machines (GBM, Fig. 3.) and Artificial Neural Networks (Fig. 4.), in addition to their assemblies, were proposed to use in this experiment in order to avoid over-training in the models and to be able to obtain more reliable comparative results [2-9]. The first step to the experiment was to preprocess the data obtained from the robotic arms; then, this preprocessed data was fed into the respective algorithms and ensembles following the methodology proposed in the figures 7 and 8. Finally, the results were statistically compared in order to point out the best results.

1.1 Hypothesis

From the processing of the information obtained from robotic arms for palletizing through various Machine Learning algorithms, it is possible to develop an intelligent

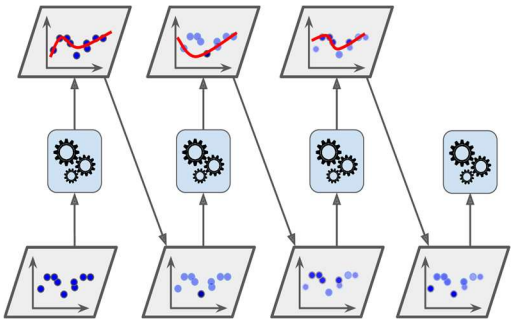


Fig. 3. Visual representation of the training of a GBM algorithm (Géron, 2019).

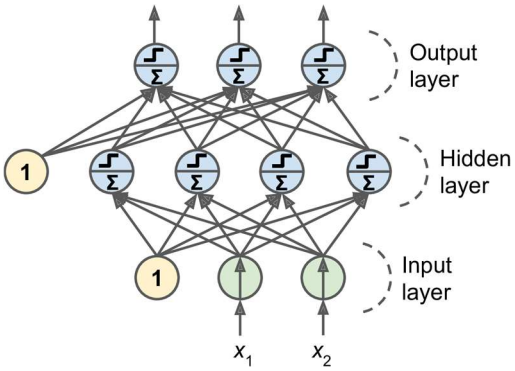


Fig. 4. Visual representation of an artificial neural network (Géron, 2019).

preventive diagnosis system for these devices.

1.2 Objectives

- Compare at least three Machine Learning algorithms for their application in data science with information from robotic arms for palletizing.
- Study the packages proposed to aid in the application of the algorithms.
- Perform the analysis and pretreatment of data obtained from robotic arms for palletizing.
- Implement data mining for use with Machine Learning.
- Analyze and apply machine-learning meta-algorithms, in addition to at least three metrics for the evaluation of algorithms in the prediction and classification of failures in robotic arms for palletizing.
- Compare the results obtained with the applied algorithms.

2 Data Preprocessing

The robot's user interface (Fig. 5.) offers a historical report of its operation that is displayed in seven columns: Timestamp, Date, Time, Error Source (12 possibilities),

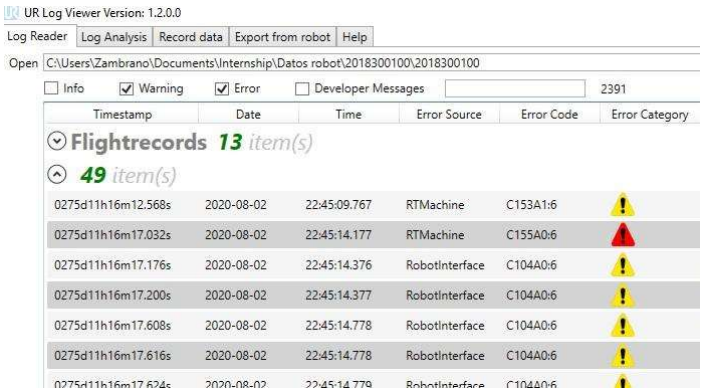


Fig. 5. Screenshot of the UR report visualization tool user interface.

Error Code (79 possibilities), Error Category (3 possibilities) and Description. By extracting the data from said interface, a text file (Fig. 6.) is obtained with the historical report of the robotic arm operation.

The resultant data set contained 21800 observations, to form an historical report of the robot's operation until its definitive failure. In order to consider preventive maintenance for a robotic system such as this robot, there must be considerations about the error messages and their relationships with the sources of the errors, and their categories.

Therefore, it was very important for this experiment to work with relevant information and preprocess the data to avoid using null values or empty strings that could alter the necessary predictions for a system to be of used as preventive maintenance to avoid failures and time out of service.

2.1 Tools Used for the Experiments

The experiments were carried out with the programming languages R[2] and Python[3] on the platforms R Studio[4] and Google Colaboratory[5], with the intention of comparing the results using tools that are freely available for their use by any other researcher. Within the previously mentioned development environments, the packages and libraries e1071, gbm, neuralnet, Scikit-Learn and AutoML were used for these experiments.

2.2 Experimentation Procedure

The following images (Figures 7 and 8) show a flow diagram of the experimentation process to follow. It is very important to emphasize that this is an iterative process in which the preprocessing of the data plays a fundamental role. This is because if an

[2] Available at: https://www.r-project.org/
[3] Available at: https://www.python.org/
[4] Available at: https://www.rstudio.com/
[5] Available at: https://colab.research.google.com/notebooks/intro.ipynb

log_history - Notepad

File Edit Format View Help

```
3.5 :: 0275d11h16m12.568s :: 2020-08-02 22:45:09.767 :: -3 :: C153A1:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m12.576s :: 2020-08-02 22:45:09.769 :: -3 :: C153A1:5 :: 3 :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m13.688s :: 2020-08-02 22:45:10.869 :: -2 :: C0A0:5 :: 1 :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m17.032s :: 2020-08-02 22:45:14.177 :: -3 :: C155A0:6 :: null :: 4 ::  ::  :: 0
3.5 :: 0275d11h16m17.032s :: 2020-08-02 22:45:14.178 :: -3 :: C155A0:5 :: 9 :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m17.064s :: 2020-08-02 22:45:14.276 :: -2 :: C100A4:6 :: null :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m17.176s :: 2020-08-02 22:45:14.376 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.200s :: 2020-08-02 22:45:14.377 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.608s :: 2020-08-02 22:45:14.777 :: -2 :: C100A3:6 :: null :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m17.608s :: 2020-08-02 22:45:14.778 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.616s :: 2020-08-02 22:45:14.778 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.624s :: 2020-08-02 22:45:14.779 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.632s :: 2020-08-02 22:45:14.779 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.640s :: 2020-08-02 22:45:14.779 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.648s :: 2020-08-02 22:45:14.779 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.656s :: 2020-08-02 22:45:14.884 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m17.664s :: 2020-08-02 22:45:14.884 :: -2 :: C104A0:6 :: null :: 2 ::  ::  :: 0
3.5 :: 0275d11h16m21.496s :: 2020-08-02 22:45:18.685 :: -2 :: C0A0:5 :: 1 :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m58.288s :: 2020-08-02 22:45:20.492 :: -2 :: C100A1:6 :: null :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m58.288s :: 2020-08-02 22:45:20.492 :: -2 :: C101A0:6 :: null :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m58.287s :: 2020-08-02 22:45:20.492 :: 30 :: C50A83:6 :: null :: 1 ::  ::  :: 0
3.5 :: 0275d11h16m58.344s :: 2020-08-02 22:45:20.493 :: -2 :: C0A0:12 :: null :: 1 :: URSafetyA 3.5
3.5 :: 0275d08h00m21.887s :: 2020-08-02 22:45:20.535 :: -5 :: C0A0:7 :: null :: 1 ::  :: La suma de
3.5 :: 0275d08h00m21.887s :: 2020-08-02 22:45:20.558 :: -5 :: C0A0:7 :: null :: 1 ::  :: La suma de
```

Fig. 6. Screenshot of the data offered by the historical report of the robot in a text file.

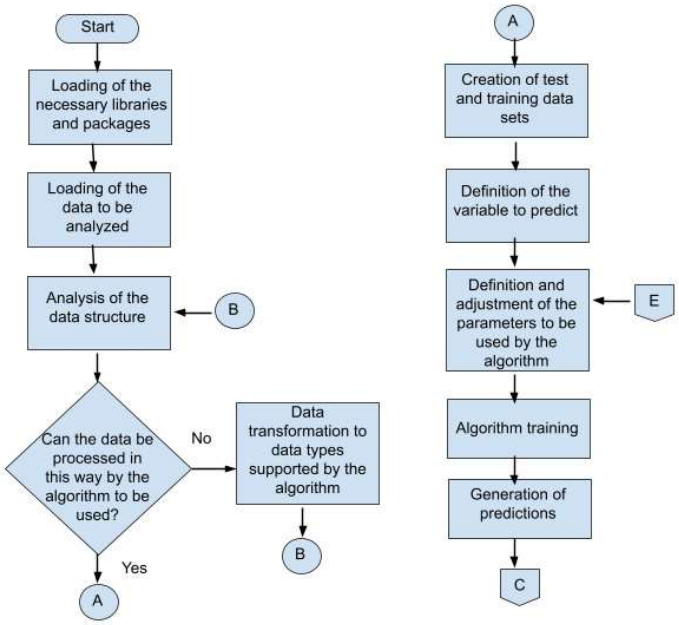


Fig. 7. Flow diagram of the experimentation process (1).

algorithm is fed with irrelevant data such as empty values, the results offered by this Machine Learning system will not be relevant either.

3 Comparative Analysis of Results

The five common metrics used to evaluate the results of the SVM (R, e1071), Gradient Boosting Machine (R, gbm), RNA - Multilayer Perceptron (R, neuralnet) and Ensemble (Stack) - SVR, GBR and MLP (Python, Scikit-Learn) models, are as follows:

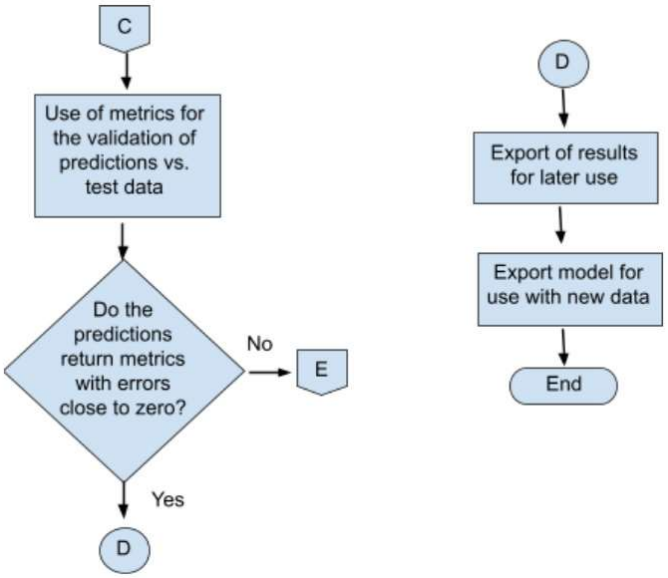


Fig. 8. Flow diagram of the experimentation process (2).

– Mean absolute percentage error (MAPE),

– Root mean squared logarithmic error (MSLE),

– Mean absolute error (MAE),

– Mean square error (MSE),

– Root mean square error (RMSE).

In the case of the experiment carried out with the AutoML ensemble algorithm – including StackedEnsemble, XGBoost, GBM, DRF, DeepLearning and XRT (Python, H2O), the three metrics in common with the other experiments are as follows:

– Mean absolute error (MAE),

– Mean square error (MSE),

– Root mean square error (RMSE).

It is important to emphasize that when using each of the errors described above as evaluation metrics for a regression model (as has been done in this case), the objective is that the value of the error obtained is as close to zero as possible. On the other hand, a value that is away from zero denotes a greater error in the predicted values.
With the intention of standardizing the performance metrics for this model, a comparative table (Table 1) was created in order to consolidate and compare the results, following the example provided by Borja-Robalino et. al. in 2020. In addition to this

Table 1. Comparison of evaluation metrics returned by the analyzed algorithms.

Model	MAPE	MSLE	MAE	MSE	RMSE
SVM (R, e1071)	0.0231	0.0077	0.0451	0.07349	0.2711
Boosting (R, gbm)	0.0889	Inf	0.1059	0.24652	0.4965
RNA Multilayer Perceptron (R, neuralnet)	0.9444	0.1919	1.1857	103.304	10.1639
Ensemble - SVR, GBR and MLP (Python, Scikit-Learn)	**0.0006**	**0.0001**	**0.0001**	0.00183	0.04282
AutoML (Python, H2O)	N/A	N/A	0.0011	**0.00065**	**0.00065**

previous step, a graph (Fig. 9.) was also created as a visual support to identify the difference in the obtained errors, in order to meet the objectives of this experiment.

4 Conclusions and Future Work

4.1 Conclusions

Machine Learning algorithms in ensembles have returned better results for this experiment than the use of algorithms independently. Despite not returning the best results in all of the analyzed metrics, the model developed with AutoML is much easier to generate than the model developed with Scikit-Learn from independent models.

The model generated with AutoML is able to return evaluation metrics in a simpler way compared to the other models, since it shows a set of them automatically, while in the other experiments, the metrics had to be explicitly required one by one. Due to the performance of the model generated from an ensemble of Machine Learning algorithms with H2O's AutoML tool, its practical use in an intelligent preventive maintenance system has been positively validated.

4.2 Future Work

The model generated during this experiment from the use of H2O's open-source tool AutoML has the possibility of being hosted within an online platform or within a desktop application. For this reason, with a little extra work it is possible to develop an intelligent preventive maintenance system for the robotic arms of the brand Universal Robots.

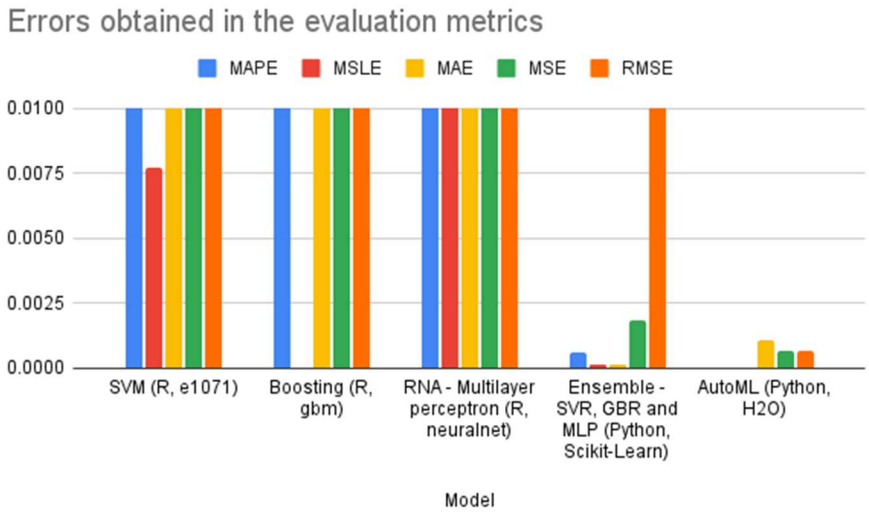


Fig. 9. Comparative graph of the errors obtained in the evaluation metrics of each used algorithm.

In the same way, the development of this work opens the door to the possibility of applying the methodology for data preprocessing and the use of the algorithm assemblies that were shown in this document with data from robotic arms of other brands. The key point with respect to the above will be to properly identify the information offered by other robotic arms and the adaptation of the proposed methodology, for later use within an intelligent system that includes the ensembles of Machine Learning algorithms used in this work.

References

1. Thomas, D. S.: The costs and benefits of advanced maintenance in manufacturing. National Institute of Standards and Technology, pp. 1–34 (2018) doi: 10.6028/NIS T.AMS.100-18
2. Bhandari, H.: Data analytics to support predictive maintenance (2020)
3. Borgi, T., Hidri, A., Neef, B., Naceur, M. S.: Data analytics for predictive maintenance of industrial robots. In: Proceedings of International Conference on Advanced Systems and Electric Technologies, pp. 412–417 (2017) doi:10.11 09/ASET. 2017.7983729
4. Çinar, Z. M., Nuhu, A. A., Zeeshan, Q., Korhan, O., Asmael, M., Safaei, B.: Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. Sustainability (Switzerland), vol. 12, no. 19 (2020) doi: 10.3390/su12198211
5. Panicucci, S., Nikolakis, N., Cerquitelli, T., Ventura, F., Proto, S., Macii, E., Makris, S., Bowden, D., Becker, P., O'mahony, N., Morabito, L., Napiore, C., Marguglio, A., Coppo, G., Andolina, S.: A cloud-to-edge approach to support predictive analytics in robotics industry. Electronics (Switzerland), vol. 9, no. 3, pp. 492 (2020) doi: 10.3390 /electronics9030492
6. Xiang, S., Huang, D., Li, X.: A generalized predictive framework for data driven prognostics and diagnostics using machine logs. In: Proceeding of IEEE Region 10 Annual International

Conference, Proceedings/TENCON, pp. 695–700 (2018) doi: 10.1109/TENCON.2018.8650152
7. Cook, D.: Practical machine learning with H2O. O'Reilly Media, Inc, (2017)
8. Géron, A.: Hands-On machine learning with Scikit-Learn, Keras, and Tensor Flow. 2nd Edition. O'Reilly Media, Inc (2019). https://www.oreilly.com/library/ view/hands-on-machine-learning/9781492032632/
9. Borja-Robalino, R., Monleón-Getino, A., Rodellar, J.: Estandarización de métricas de rendimiento para clasificadores machine y Deep Learning. (2020)
10. H2O. Retrieved July 13, 2021, from https://www.h2o.ai/products/h2o/

ISSN 1870-4069

Spiking Neural Network Implementation of LQR Control on an Underactuated System

Jorge A. Juárez-Lora, Humberto Sossa, Victor H. Ponce-Ponce,
Elsa Rubio-Espino, Ricardo Barrón-Fernández

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{jjuarezl2020, hsossa, vponce, erubio, rbarron}@cic.ipn.mx

Abstract. This writing presents an architecture proposal designed to implement a control loop in a mobile-wheeled under-actuated inverted pendulum system, using spiking neural networks, linear quadratic regulator control technique, and a neural framework that allows us to define the neuron ensembles specification to represent specific control signals.

Keywords: Robotics, neural networks, spiking neural networks, machine learning, neurorobotics, neurocomputing.

1 Introduction

4.0 Industry has brought a massive proliferation of sensors and data acquisition devices for monitoring and analysis purposes. This situation has escalated quickly, as the amount of recollected data overpasses any computation and storage capacities needed to provide information solutions that allow intelligent decision-making using *Big Data* process techniques. Storing all the information is not feasible anymore, so new analysis techniques have appeared, such as artificial neural networks, which have proven to be very useful in online learning scenarios.

Spiking neural networks (SNN), also known as artificial third-generation neural networks, intend to emulate biological plausibility, physical, chemical, and biological mechanisms, allowing computation and Hebbian learning to occur in biological living systems. These models have optimal characteristics for hardware implementation [1, 2].

It promises enormous parallel computing capacity and low energetic usage, enabling feasible online learning platforms to implement size and power restriction applications, such as robotics. Neurorobotics [3, 4] is a discipline that takes as a challenge to design control mechanisms, hardware, and implementation techniques for robotic applications. These agents adapt their behavior up to changes in themselves or the environment dynamics.

This situation can be seen in biological systems with growing limbs or holding a heavy object, alien to its composition, or perhaps, aging, which modifies friction in arm or leg joints. One of the most evident obstacles in neurorobotics development is the shortage of physical platforms for neuromorphic computation. However, there are

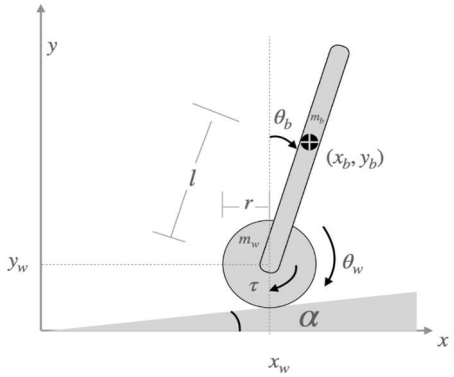


Fig. 1. 2D MWIP model, extracted from [17].

simulation platforms that allow designing the neural structures needed for future implementations.

Some efforts such as *Human Brain Project* [5], which has a neurorobotics platform [6], allow to simulate a brain or physical body and explore how to control movement, stimuli reaction and learn from a virtual or real environment. Another platform, primarily focused on SNN implementation, is called Nengo [7-9], a tool that allows to build and design SNNs architectures. It is quite flexible, as the user can define its neural models, its own learning rules, optimization methods, reuse of subnetworks, data input, and even has libraries for exporting these models to neuromorphic hardware or FPGA implementations.

As a small tour in literature, in [10], an adaptive control method proposed in [11] is used, allowing them to control a three-link arm in simulation, using a spiking neural network structure designed to estimate the inverse jacobian dynamics. Here, authors name part of their proposed neural network as their biological counterpart to match specific tasks made by natural brains. In [12], control of a simulated robotic arm, without path planning, is achieved using SNNs and *motor primitives*. In [13], a biologically inspired spiking neural network (SNN) for soft-grasping to control a robotic hand, used for robots interacting with objects shaped for humans, is presented.

Finally, in [14], a hardware adaptive control implementation of a Kinova Jaco robotic arm using the Loihi platform [15] is introduced. These three examples have something in common; these are complete actuated systems. In this work, an implementation of a linear quadratic regulator (LQR) strategy using SNN is presented as an introductory example of how Nengo and SNN can be used for under-actuated systems, showing which obstacles must be tackled to perform precise control signal representation.

The Mobile Wheeled Inverted Pendulum (MWIP) is an easily controllable system for a human with a bit of practice but a challenge in control theory.

Although some of these under-actuated systems show controllability under linearization around a certain equilibrium point, the control tasks entitle arbitrary output reference trajectory tracking, taking the system state away from the equilibrium point, thus overcoming a traditional obstacle to linearization-based control of nonlinear systems [16].

This document is organized as follows. Section 2 describes the MWIP dynamics and the LQR control technique used for controlling the system. Section 3 describes which methodology was used to create an SNN structure to represent the LQR controller and shows its implementation in Nengo software simulation. Section 4 shows the configuration parameter for the simulation and its results. Finally, section 5 is used for conclusions and proposed future work.

2 Dynamics and Control Strategy of the Robotic System

2.1 Dynamic Model of the System

Fig. 1 shows the graphical representation of the MWIP (Mobile Wheeled Inverted Pendulum). Here, x_w, y_w are the wheel coordinates, x_b, y_b are the mass center coordinates of the bar, α is the plane's angle inclination, m_b, m_w stand for the bar and the wheel's mass, respectively, I_b, I_w are the moments of inertia from the bar and the wheel, L is the bar's length, r is the radius of the wheel, and θ_w, θ_b are the states of the system, which stand for the rotation angle of the wheel, and the bar's inclination, in that order.

The robotic system corresponds to a second-order underactuated system [18]. Starting from the modeling dynamics using Euler-Lagrange technique in [17], setting $\alpha = 0$ leads to a system with the following depiction:

$$M(q)\ddot{q} = C(q, \dot{q}, u), \tag{1}$$

where: $M(\cdot)$ is the inertia matrix, $\mathrm{C}(\cdot)$ groups *the coriolis, gravity, and control terms,* and the extended form of this equation is:

$$\begin{pmatrix} (m_b + m_w)r^2 + I_w & m_b L r\, cos(\theta_b) \\ m_b L r\, \cos(\theta_b) & m_b L^2 + I_b \end{pmatrix} \begin{pmatrix} \ddot{\theta_w} \\ \ddot{\theta_b} \end{pmatrix} = \begin{pmatrix} u + m_b L r\, \dot{\theta_b}^2\, \sin(\theta_b) \\ -u + m_b g\, L\, \sin(\theta_b) \end{pmatrix}. \tag{2}$$

Here, u is the control signal. It is worth mentioning the motor is mounted in the hub that connects the wheel and bar, so the used torque is the same but in the opposite direction. To obtain the accelerations of the system, we rewrite eq. (2) as:

$$\ddot{q} = M(q)^{-1} \cdot c(\dot{q}, q). \tag{3}$$

Both second-order differential equations can be represented in four first-order equations, this is, rewriting the system in a space state manner, such as $\dot{x} = f(q, \dot{q}, u)$. Setting $x = [x_1, x_2, x_3, x_4] = [\theta_w, \dot{\theta_w}, \theta_b, \dot{\theta_b}]$, the system can be linearized in its equilibrium states $\theta_b = \dot{\theta_w} = \dot{\theta_b} = 0$, which is equivalent to a pendulum in an upright position. Therefore, the linearized system results in the form $\dot{x} = Ax + Bu$ with the matrices A, B given by:

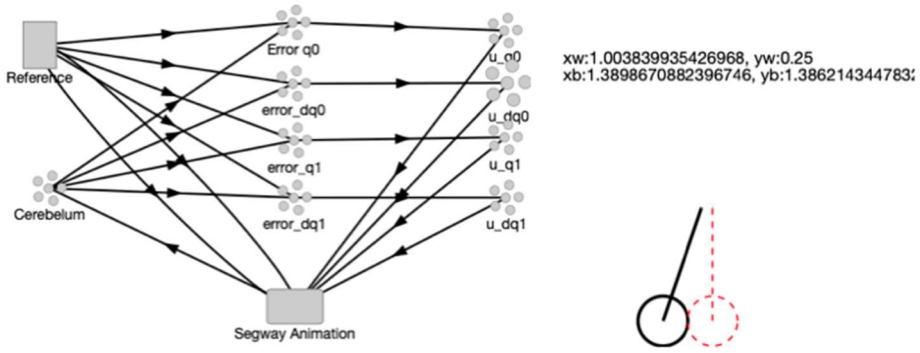


Fig. 2. SNN structure proposed for the control simulation.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-gL^2m_b^2r}{z} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{gLm_b^2r^2 + gLm_wm_br^2 + I_wgLm_b}{z} & 0 \end{bmatrix}, \tag{4}$$

$$B = \begin{bmatrix} 0 \\ \frac{m_bL^2 - m_brL + I_b}{z} \\ 0 \\ -\frac{I_w + m_br^2 + m_wr^2 - Lm_br}{z} \end{bmatrix}. \tag{5}$$

With:

$$z = I_bI_w + I_wL^2m_b + I_bm_br^2 + I_bm_wr^2 + L^2m_bm_wr^2. \tag{6}$$

It can be easily shown that the system is fully controllable and observable.

2.2 Linear Quadratic Regulator (LQR)

The goal then is to move the vector state x from an initial condition to a desired vector x_d. Finding a control law $u = -K_r(x - x_d)$ for the MWIP system, which moves the closed-loop eigenvalues of the system as far as required on the left half of the complex plane, in order to achieve optimal control, is a task of optimization.

Choosing over stable eigenvalues might cause the system to overreact to small perturbations or noise or require high control signals, which might overpass the actuator's capacity. On the other side, choosing eigenvalues as close as possible to the right half in the complex plane might result in long stabilization times and small control signals, leading to instability.

The Linear Quadratic Regulator [19] (LQR hereinafter) consists in to deliver a full-state feedback control method that minimizes the following cost function:

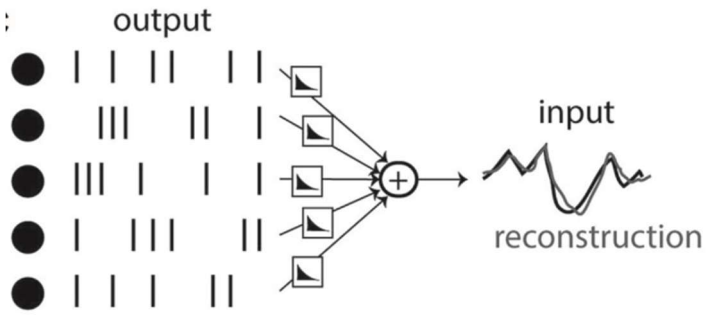


Fig. 3. Spike-based sparse coding. A reconstruction of the signal is obtained from combining filtered spike trains together, and spikes are timed to make the reconstruction accurate. Extracted from [26].

Table 1. MWIP model parameters used for simulation.

Parameter	Value
m_b	$1[kg]$
m_w	$2[kg]$
L	$1.2[m]$
r	$0.25[m]$
I_w	$10[N/m]$
I_b	$10[N/m]$

$$J(t) = \left(\int_0^\infty (x - x_d)^T Q (x - x_d) + u^T R u \right) dt. \tag{4}$$

The function in (4) pictures a balance between the energetic cost of an effective state regulation, which is intended to be minimized, and a quicker control response, which is intended to be fast.

These objectives are regulated by $Q = Q^T \geq 0$ and $R = R^T \geq 0$ respectively, and they can be selected as wished to prioritize control objectives. As bigger is Q, it will move the system to the desired vector state as soon as possible. As big as R might be, lower control signals will be the priority, while $J = \lim_{t \to \infty} x(t) = 0$.

As $J(\cdot)$ is a quadratic function, there exists an analytic solution for control weights in K_r given by:

$$K_r = R^{-1} B P, \tag{5}$$

where P is the Ricatti's algebraic equation solution:

$$PA + AP^T - PBR^{-1}B^T P + Q = 0. \tag{6}$$

In order to solve eq. (6) there are several software-implemented methods [20, 21], which start from a known A, B for a $\ddot{x} = f(q, \dot{q}, u)$ system dynamics.

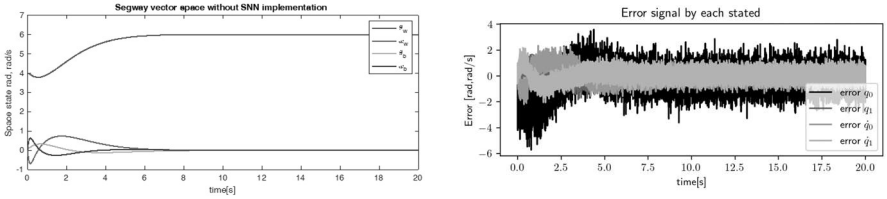


Fig. 4. Evolution of the Vector State (left: simple control loop simulation without SNN, right: using the proposed SNN structure).

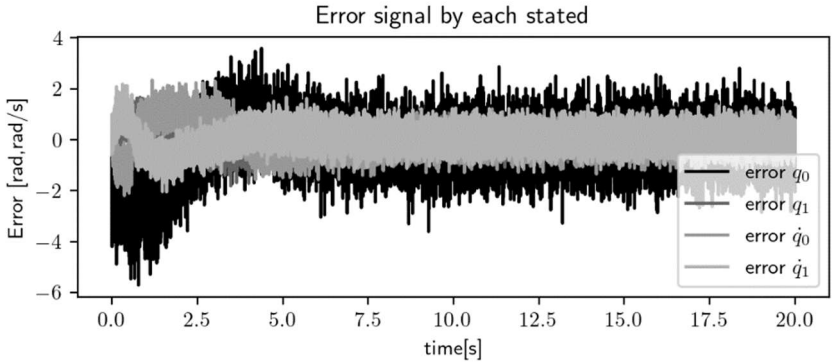


Fig. 5. Evolution of error signal for each state, simulating the proposed SNN structure.

3 Neural Network Simulation

In order to design and implement the neural network, the principles developed in *Neural Engineering Framework (NEF)* [27, 28] are used. NEF can be seen as a *'neural compiler'* that guarantees an optimal global approximation of the defined dynamic equations by the user-defined groups of neurons called ensembles. Given a signal, it is possible to use a nonlinear encoding matrix E to parse it into a spike domain.

Then, recover an approximation of the original signal through a linear decoding matrix D, obtaining then the synaptic weights $W = ED$ for the ensemble. Once connected, the resulting network approximates the ideal signal according to neural heterogeneity, stochasticity, and connectivity, affecting its performance.

This is called Spike-based sparse coding, and it can be seen in Fig. 3 [26] Fig. 2 shows the implemented SNN using the simulation software Nengo [22], which provides libraries for define and connect ensembles, which once connected between them, Nengo will find the appropriate synaptic weights using a learning rule (such as STDP) to represent any stimuli value proportioned, and represent it on the next ensemble.

This is achieved using the methodology described in NEF and BCM theory [23,24], which describes how synaptic plasticity on cortical neurons is stabilized by the average postsynaptic activity. All the neurons in this proposed model use *Leaky Integrate and Fire (LIF)* [25] model for its representation.

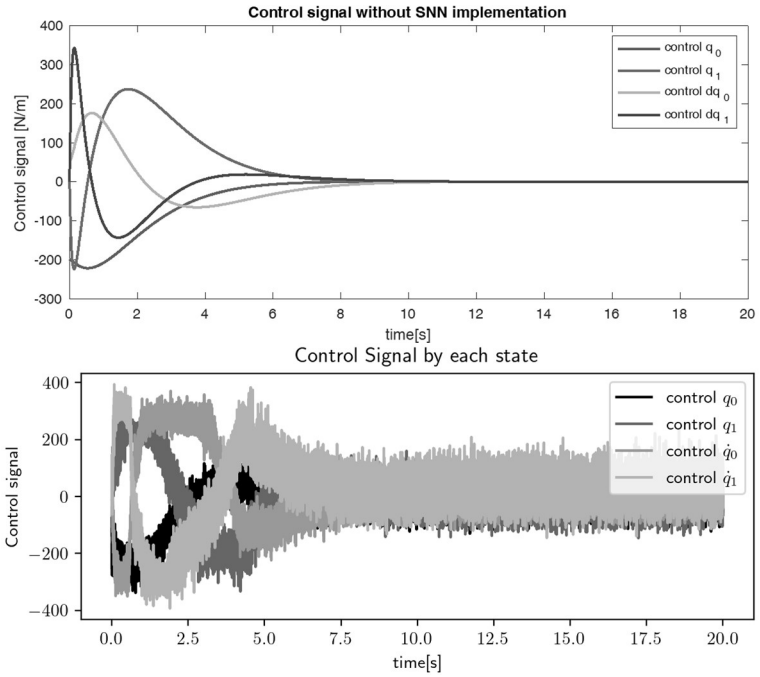


Fig. 6. Evolution of control signal for each state (Up, simple control loop simulation without SNN, down, simulating the proposed SNN structure).

As part of the encoding, each ensemble can encode a signal from a minimum to a maximum value. Nengo allows the user to modify this parameter in the ensemble construction, in order to define the represented function domain. This domain is called radius. By default, this radius is set between [-1, 1].

However, as soon we reset to a bigger domain, the number of neurons in the ensemble needs to be implemented. If this is not the case, the output signal will lose resolution, creating noisy output signals. A small radius then implies better precision. Up next, each of the elements of the proposed structure in Fig. 2 is described:

– Reference: A small function that returns the desired vector state x_d.
– Cerebellum: 1000×4 neuron ensemble, which encodes the state from the MWIP. Radius (-10, 10). Similarly called like this as [10].
– Error:100×4 neuron ensemble encodes the signal reference and computes the difference between the actual and the reference vector state. Radius (-3.1416, 3.1416).
– Control U: 1000×4 neuron ensemble, which computes the control signal for each state variable. Range (-350,350).
– MWIP Animation: Node used for MWIP simulation, with one control input (u) and four outputs $(\theta_w, \dot{\theta_w}, \theta_b, \dot{\theta_b})$.

The ensemble's radius was selected according to the function domain for each state variable represented. Nevertheless, some states such as angular velocities $\dot{\theta_w}$, $\dot{\theta_b}$ have

virtually open domains, while in practice, these are limited. Another example is the computed control signal. Appropriate radii were selected to achieve the maximal values required for the given initial conditions.

4 Simulation Scenario

4.1 Simulation Parameters

In order to evaluate the architecture performance, the MWIP starts from an initial state $x_i = [4,0,0.1,0]$, with a desired vector state $x_d = [6,0,0,0]$. Table 1 shows the MWIP model parameters used and $g = 9.81m/s^2$. For the control loop, $u = -K(x - x_d)$, setting $Q = I$ and $R = 0.001$, the matrix $K = [-100, -323.3434, -542.0927, -541.08]$, using the described methods in [20, 21]. The simulation period has a duration of $t = 20s$ with a step integration of $1\,ms$.

4.2 Results

Fig. 4 shows the MWIP space state evolution. We can see the initial values evolve towards the desired vector state successfully, with a smooth transition and finishing with a relatively small error, as shown in Fig. 5. It is worth mentioning the stochastic and noisy nature of the control signals (See Fig. 6), due to spike-based sparse coding, oscillate but achieve to represent the desired control signal.

5 Conclusions and Future Work

During the development of this work, a neural architecture based in a biologically plausible model can be used to emulate system dynamics and control implementation. It has been shown how to declare control structures and implement them successfully in under-actuated robotic systems. It is shown that a correct radius specification for each ensemble reflects the precision of its output control signal.

However, while this control strategy reaches a desired vector state, the produced control signals seem to be noisy and stochastic. It reminds those produced by control strategies such as sliding modes [17]. While noise added by the neural dynamics might be problematic, it adds a small value, avoiding singular matrices during encoding and decoding processes used in NEF [7].

This also might be beneficial to prevent an overfitting case. We consider possible future works, like a *Kalman* filter implementation in SNN networks for signal cleaning, or online dynamics estimation of the plant to compute the control signal, exploring other control techniques such as ADRC [29] for unknown plant dynamics and its possible neuromorphic implementation in FPGA or ASIC devices.

Acknowledgments. This work was supported by the Instituto Politécnico Nacional (IPN) and Secretaría de Investigación y Posgrado (SIP-IPN) under the projects: SIP20200630, SIP20201397, SIP20200885, SIP20210788, SIP20210124, and SIP20211657, Comisión de Operación y Fomento de Actividades Académicas

(COFAA-IPN), also the Consejo Nacional de Ciencia y Tecnología (CONACYT-México) under projects 65 (Frontiers of Science 2015) and 6005 (FORDECYT-PRONACES). Juárez-Lora would like to thank to CONACYT for the grant proportioned for his PhD studies.

References

1. Yan, Y., Stewart, T., Choo, X., Vogginger, B., Partzsch, J., Hoeppner, S., Kelber, F., Eliasmith, C., Furber, S., Mayr, C.: Comparing Loihi with a SpiNNaker 2 prototype on low-latency keyword spotting and adaptive robotic control. Neuromorphic Computing and Engineering, vol. 1, no. 1 (2021) doi: 10.1088/ 2634-4386/abf150
2. Lobo, J. L., Del Ser, J., Bifet, A., Kasabov, N.: Spiking neural networks and online learning: An overview and perspectives. Neural Networks, vol. 121, pp. 88–100 (2020) doi: 10.1016/j.neunet.2019.09.004
3. Galindo, S., Toharia, P., Robles, Ó., Ros, E., Pastor, L., Garrido, J.: Simulation, visualization and analysis tools for pattern recognition assessment with spiking neuronal networks. Neurocomputing. vol. 400, pp. 309–321 (2020)
4. Bogdan, P., Marcinnò, B., Casellato, C., Casali, S., Rowley, A., Hopkins, M., Leporati, F., D'Angelo, E., Rhodes, O.: Towards a bio-inspired real-time neuromorphic cerebellum. Frontiers in Cellular Neuroscience, vol. 15, pp. 130 (2021)
5. Amunts, K., Knoll, A. C., Lippert, T., Pennartz, C., Ryvlin, P., Destexhe, A., Jirsa, V. K., D'Angelo, E., Bjaalie, J. G.: The human brain project-synergy between neuroscience, computing, informatics, and brain-inspired technologies. Public Library of Science Biology, vol. 17, no. 7 (2019) doi: 10.1371/journal .pbio.3000344
6. Falotico, E., Vannucci, L., Ambrosano, A., Albanese, U., Ulbrich, S., Vasquez-Tieck, J. C., Hinkel, G., Kaiser, J., Peric, I., Denninger, O., Cauli, N., Kirtay, M., Roennau, A., Klinker, G., von Armin, A., Guyot, L., Peppicelli, D., Martínez-Cañada, P., Ros, E., Maiers, P.: Connecting artificial brains to robots in a comprehensive simulation framework: The neurorobotics platform. Frontiers in Neurorobotics, vol. 11, no. 2 (2017) doi: 10.3389/fnbot.2017.00002
7. Voelker A. R., Eliasmith C.: Programming neuromorphics using the neural engineering framework. In: Thakor N.V. (eds) Handbook of Neuroengineering (2021) doi: 10.1007/978-981-15-2848-4_115-1
8. Rasmussen, D.: NengoDL: Combining deep learning and neuromorphic modelling methods. Neuroinformatics. vol. 17 (2019) doi: 10.1007/s12021-019-09424-z
9. The Nengo neural simulator-Nengo. Available at https://www.nengo.ai/
10. DeWolf, T., Stewart, T., Slotine, J. J., Eliasmith, C.: A spiking neural model of adaptive arm control. In: Proceedings of the Royal Society: Biological Sciences, vol. 283, no. 1843 (2016) doi: 10.1098/rspb.2016.2134
11. Cheah, C. C., Liu, C., Slotine, J. J.: Adaptive tracking control for robots with unknown kinematic and dynamic properties. The International Journal of Robotics Research, vol. 25, no. 3, pp. 283–296 (2006)
12. Tieck, J., Steffen, L., Kaiser, J., Roennau, A., Dillmann, R.: Controlling a robot arm for target reaching without planning using spiking neurons. In: IEEE 17th International Conference on Cognitive Informatics Cognitive Computing, pp. 111–116 (2018) doi: 10.1109/ICCI-CC.2018.8482049
13. Tieck, J., Secker, K., Kaiser, J., Roennau, A., Dillmann, R.: Soft-Grasping with an anthropomorphic robotic hand using spiking neurons. Robotics and Automation Letters, vol. 6, no. 2, pp. 2894–2901 (2021) doi: 10.1109/LRA.20 20.3034067

14. DeWolf, T., Jaworski, P., Eliasmith, C.: Nengo and Low-Power AI Hardware for Robust, Embedded Neurorobotics. Frontiers in Neurorobotics, vol. 14. (2020) doi: 10.3389/fnbot.2020.568359
15. Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G., Joshi, P., Plank, P., Risbud, S.: Advancing neuromorphic computing with Loihi: A survey of results and Outlook. In: Proceedings of the IEEE, vol. 109, no. 5, vol. 911–934 (2021) doi: 10.1109/JPROC.2021.3067593
16. Sira-Ramírez, H.: Active disturbance rejection control of dynamic systems: a flatness-based approach. Oxford: Butterworth-Heinemann (2017)
17. Ri, S., Huang, J., Wang, Y., Kim, M., An, S.: Terminal sliding mode control of mobile wheeled inverted pendulum system with nonlinear disturbance observer. Mathematical Problems in Engineering, vol. 2014, pp. 1–8 (2014) doi: 10.1155/2014/284216
18. Krafes, S., Zakaria, C., Saka, A.: A Review on the control of second order underactuated mechanical systems. Complexity, pp. 1–17 (2018) doi: 10.1155/20 18/9573514
19. Brunton, S., Kutz, J.: Linear control theory. Data-Driven Science and Engineering, Cambridge University Press, pp. 276–320 (2019) doi: 10.1017/9781108380 690.009
20. Benner, P., Li, J. R., Penzl, T.: Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. Numerical Linear Algebra with Applications, vol. 15, no. 9, pp. 755–777 (2008)
21. Arnold, W. F., Laub, A.: Generalized eigen problems algorithms and software for algebraic Ricatti equations. In: Proceedings of the IEEE, vol. 72, no. 12, pp. 1746–1754 (1984)
22. Rasmussen, D. NengoDL: Combining deep learning and neuromorphic modelling methods. Neuroinform vol. 17, pp. 611–628 (2019) doi: 10.1007/s120 21-019-09424-z
23. Izhikevich, E., Desai, N.: Relating STDP to BCM. Neural computation, vol. 15, no. 7, pp. 1511−1523 (2003) doi: 10.1162/089976603321891783
24. Blais B. Shouval H., Cooper L. N.: The role of presynaptic activity in monocular deprivation: Comparison of homosynaptic and heterosynaptic mechanisms. In: Proceedings of the National Academy of Sciences, vol. 96, no. 3, pp. 1083–1087 (1999)
25. Gerstner, W., Kistler, W., Naud, R. Paninski, L.: Neuronal Dynamics. from single neurons to networks and models of cognition (2006)
26. Brette, R.: Philosophy of the Spike: Rate-Based vs. Spike-Based theories of the brain. Frontiers in Systems Neuroscience, vol. 9, no. 151 (2015)
27. Eliasmith C, Anderson, C.: Neural engineering: computation, representation, and dynamics in neurobiological systems. Cambridge, MA: MIT Press (2004)
28. Eliasmith C.: How to build a brain: A neural architecture for biological cognition. Oxford University Press (2013) doi: 10.1093/acprof:oso/9780199794546.001.000
29. Aguilar-Ibanez, C., Sira-Ramirez, H., Acosta, J. Á., Suarez-Castanon, M. S.: An algebraic version of the active disturbance rejection control for second-order flat systems. International Journal of Control, vol. 94, no. 1, pp. 215–222 (2019) doi: 10.1080/00207179.2019.1589651

ISSN 1870-4069

Selection of Features for Attribution of Authorship Using a Genetic Algorithm and Support Vector Machine as a Function of Aptitude

Omar González Brito[1], José Luis Tapia Fabela[1], Silvia Salas Hernández[2]

[1] Universidad Autónoma del Estado de México, Unidad Académica Profesional Tianguistenco, Mexico

[2] Universidad Autónoma del Estado de México, Centro Universitario UAEM Atlacomulco, Mexico

{gonzalezbritoomar, joseluis.fabela, salashernandezsilvia}@gmail.com

Abstract. In the present investigation, a genetic algorithm was developed with a support vector machine as an aptitude function. This algorithm has the purpose of searching vector sub-spaces of features that improve the authorship attribution. This method reduces dimensionality, improves classification, and identifies which are the most significant features to differentiate between authors. To determine the author, it is necessary to analyze linguistic features or textual features that allow us to find the author's writing style. In the experiments carried out, it is observed that the results obtained are satisfactory, where they proposed six types of samples, of which three were balanced and three were imbalanced, where the samples of size five improved the accuracy by 14.0%, in the sample of size 10 the accuracy is improved by 5.3% and in the sample size 10:20 the accuracy is improved by 6.15%.

Keywords: Genetic algorithm, selection of features, support vector machine.

1 Introduction

At present, authorship analysis has become a major problem in many areas; among them, information retrieval, computational linguistics, and forensic linguistics. A great diversity of corpus has been created, covering different contexts such as; emails, journalistic notes, literary works, and social networks [1, 2, 3, 4, 5, 6]. The authorship attribution consists of identifying the author of a text within a set of candidates through their textual features that allow differentiating the writing style of an author [7, 8]. The computational problem of authorship attribution has been addressed mainly through the classification of texts [1, 8, 9, 10, 11].

In [12], it is shown that every classification task tends to have many features that can be relevant, irrelevant, and redundant. Irrelevant and redundant features degrade the performance of a classifier. This is sensitive to the selection of its features. When making the selection of features, the performance of the classifier is optimized [13], [14]. The lexical features mainly used in authorship attribution are the bag of words and n-grams models. Others found within the lexicons are the original words and the empty words, the count of the length of the sentences or the number of words in the text, etc. [15, 16, 17, 18].

The main advantage of these features is domain and context independence except in languages such as Chinese. These types of features do not require any processing to obtain them or the application of pre-processing for their analysis. The selection of features is intended to eliminate irrelevant, redundant features. For the selection of features, filter methods (statistical) and wrapper methods (learning) have been implemented. This makes it possible to eliminate unnecessary features and improve the classification [19]. The filter method selects the features utilizing a relevance criterion. This criterion can be the measure of distance or dependence.

The filter methods evaluate the features independently concerning the classes of the training set. The wrapper method selects subsets of features from a classifier. The criteria for selection are according to the learning algorithm, mainly search algorithms are implemented [19, 20, 21, 22]. When characterizing a text with the n-gram model, the dimensionality of the features is very high, within this set there are relevant, irrelevant, and redundant features [12].

To choose the relevant features of an author, a genetic algorithm was implemented as a method of selection of features, which has as its main objective the search for optimal vector subspaces, which allows improving the classification [23]. In the present investigation, they were obtained better results in three out of six samples and competitive results in the other samples compared with one of the works most referenced by the state of the art present in [11].

2 Related Works

Genetic algorithms are considered meta-heuristic methods, which mimic the process of natural selection proposed by Darwin. These adaptive methods arise with the need to find solutions to problems in a complex search space. The individuals with the best performance will have the possibility of being selected to pass to the next generation and reproduce these are considered possible solutions to a particular problem [24, 25, 26, 27].

In the literature review, different works have been found for the selection of features, one of them is presented in [23], where they implement a genetic algorithm and the k-closest neighbor's classifier; select the most significant words (features) to improve accuracy in the authorship of science fiction texts. Where the dimensionality to be treated is lower compared to the analysis of n-grams at the character level. The corpus used consisted of 503 science fiction text files in English with 17 authors, using 350 for the training phase and 153 for validation.

However, in the literature review for the text classification task, better results are observed by implementing a support vector machine. Another work is presented in [14],

where he implements a genetic algorithm with a support vector machine, used for the selection of the most significant features, applied to the detection of cancer with good results. The dimensionality of the features to be selected is lower and the classes are two (it has cancer and it does not have cancer), these considerably reduce the number of features obtaining good results with only nine features.

The corpus consisted of 200 images for training and 77 for validation, the results obtained exceed 70% accuracy. The parameters used for his genetic algorithm were roulette selection, two-point crossing, Gaussian mutation, with a population of 100 individuals, and 10 generations with a mutation probability of 0.01%.

On the other hand, in [28] they implement a genetic algorithm for the selection of features, it mentions that an important part within the classification of texts is the selection of features, this type of problem seeks to optimize the process of selection of features, due to this need to implement a genetic algorithm, this type of method allows finding the best solutions to particular problems.

In this work they implement a genetic algorithm that meets the following conditions; the set of features chosen by the algorithm had to represent the category of texts, the aptitude of the individuals was evaluated individually to ensure optimal similarity between individuals, the dimension of the vector of features had to be the smallest possible size, for its evaluation used cosine similarity metrics using an elitist selection operator, generating a random population of 100 individuals and 400 generations, with a mutation rate of 0.005%.

The Corpus consisted of 1200 documents from the SOGU laboratory, of which 800 were used for training and 400 for validation, taking into account 8 categories, the results obtained show that the genetic algorithm obtains an accuracy of 84.25%, this indicates that by using genetic algorithms for the selection of features, it is possible to select the representative ones without affecting the performance of the classifier, which in this case was Naïve Bayes.

3 Method and Materials

According to the literature review, in the present investigation, a genetic algorithm was developed with a support vector machine as an aptitude function that allows selecting the most significant features using the lexical features (model of n-grams at character level) applied to the authorship attribution task to improve the classification. As observed in the state of the art, the learning method that obtains the best results in the classification is the support vector machine [11, 29].

The feature selection method that has been implemented in particular for the C10 corpus is the most frequent term filter method. This type of method selects the features according to a relevance measurement, this selection is carried out empirically. However, the wrapper methods do not perform the selection empirically, they consider the entire possible set of solutions. The best subsets of features are selected from supervised learning.

The proposed method for the selection of features consists of the development of three stages.

1. Development of a text classification method.
2. Genetic algorithm for the selection of features.

Table 1. Genetic algorithm parameters.

Poblation size	Selection	Cross	Bitwise mutation	Elitism
100 individuals	Deterministic Tournament	Uniform at 80%	0.01%	3

3. Evaluation of the features obtained with the genetic algorithm.

Text classification method: the method to be implemented consists of the following stages [31]:

- Data acquisition: the data set used in this research is available on the PAN website (https://pan.webis.de/) this corpus is used for the detection of plagiarism and currently also the authorship identification. The C10 corpus is used for the task of authorship attribution in different works of literature.
- Data analysis and labeling: In this stage, the features extraction process is carried out for each author's document. Later they are represented by the vector model.
- Construction of features and weighting: Boolean or binary weighting assigns a value of one when the term is present and a zero in the absence of the value, assigning these values in the vector representation [31, 32].
- Selection and projection of features: In the implementation with the genetic algorithm no method of selection of features is used, and for the evaluation of the features those obtained by the genetic algorithm are used.
- Training of a classification model: The learning method used is a support vector machine using a linear kernel, the parameter C equal to one. The support vector machine in the genetic algorithm is trained with the features of each of the authors, generating a model that allows an evaluation to be carried out.
- Evaluation of the solution: The metric used is accuracy, it is defined in terms of true positives (TP), False positives (FP), True negatives (TN), and False negatives (FN) as shown below [29].
- $\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN}$.

3.1 Selection of Features with the Genetic Algorithm

For the selection of features with the genetic algorithm, only the training documents are used, these are divided into a new set of training and validation, to be able to carry out the evaluation with the support vector machine. The representation used for individuals is a binary encoding, where when the value of the gene is one the feature is considered, on the other hand, if the value is zero it is discarded.

Rewriting the documents with the features to be used so that the individual can be evaluated with the support vector machine using the accuracy metric to know the individual's aptitude.

The operators used by the genetic algorithm are described below, these were determined according to the analysis presented in the experimentation section:

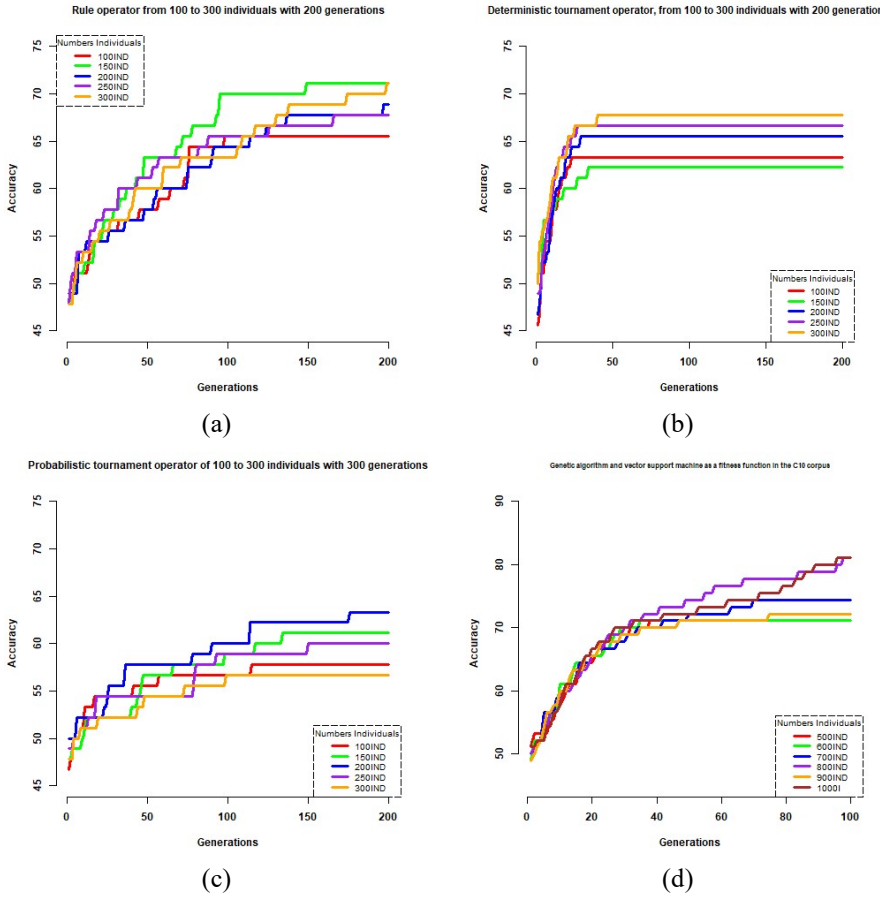


Fig. 1. Results of genetic operators.

3.2 Evaluation of the Features

The evaluation was carried out on the set of features considered the most significant by the genetic algorithm, starting from the fittest individual. The process that is carried out for the evaluation is as follows:

1. The training and validation documents are featured with the 3-gram model.
2. Boolean weighting and vector representation are performed with the features selected by the genetic algorithm.
3. The support vector machine training is performed.
4. From the model generated with the support vector machine, the validation documents are evaluated with the accuracy metric.

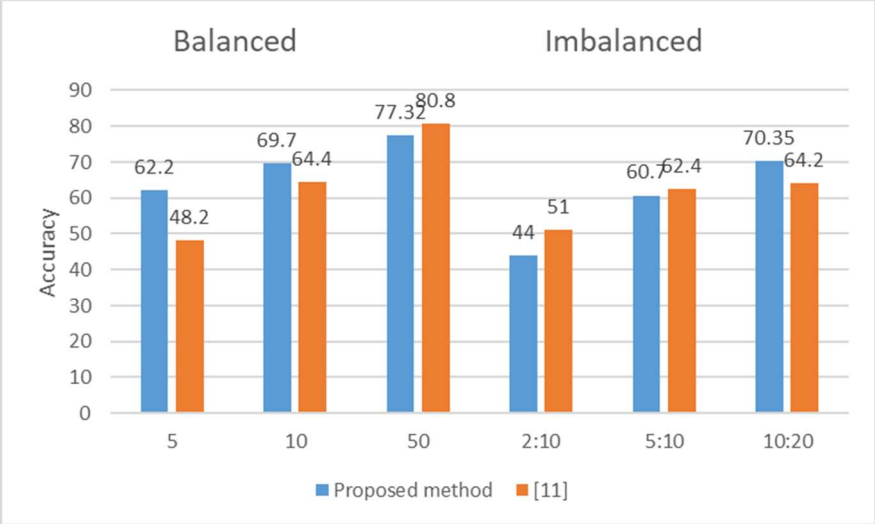


Fig. 2. Analysis of Balanced and Imbalanced samples.

4 Experimentation

The corpus used in the experimentation is the C10 corpus. An analysis of the different operators is required for a genetic algorithm with binary coding since the genetic algorithm implemented in the present investigation has an average of 5,000 genes as explained in the method each individual is a text classification process, which generates a high computational cost. Fig. 1 shows an analysis to determine the best operators for the genetic algorithm.

For the experimentation, the following parameters of the genetic algorithm were considered. Genetic operators; selection operator: roulette, deterministic and probabilistic tournament, crossover operator: uniform with 80%, the mutation operator has a probability of 0.01%. The experimentation to determine the appropriate parameters was carried out with a sample of size 10 from Corpus C10.

a. Analysis with the roulette operator, from 100 to 300 individuals with 200 generations.

For the first experiment, the following population sizes were considered; 100, 150, 200, 250, and 300 individuals (IND). As can be seen in Fig. 1a, the evolution process is slower in larger populations than in smaller populations, this is due to the fact that in smaller populations the individuals with better aptitude tend to be selected more frequently, but it is also observed in Fig. 1a shows that the genetic algorithm with this selection operator requires a greater number of generations.

b. Analysis with the deterministic tournament operator, from 100 to 300 individuals with 200 generations.

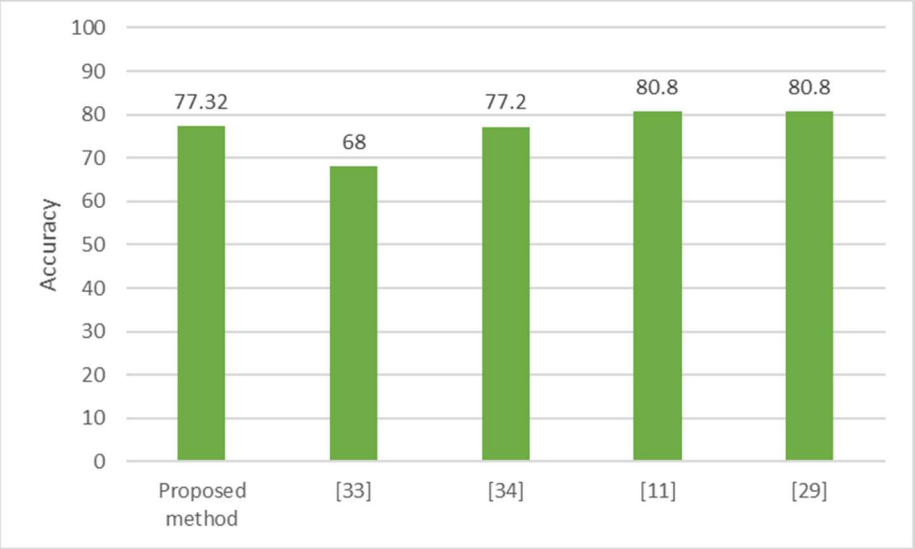


Fig. 3. Proposed method versus the state of the art.

For the second experiment, the following population sizes were considered; 100, 150, 200, 250, and 300 individuals (IND). As can be seen in Fig. 1b, the process of evolution is faster, requires fewer generations, and the greater the number of individuals in the population, the better results are obtained, because in larger populations the individuals with better aptitude tend

To be selected more frequently, according to the experimentation carried out, it is better to have more individuals than to have more generations with this operator, the computational cost is lower when the number of individuals is increased compared to the increase in generations.

c. Analysis with the probabilistic tournament operator, from 100 to 300 individuals with 200 generations. For the third experiment, the following population sizes were considered; 100, 150, 200, 250, and 300 individuals (IND). As can be seen in Fig. 1c, the smaller the

population, the better the evolution. With older populations, evolution is gradual, but it requires a greater number of generations and a greater number of individuals, which would imply a high computational cost.

d. Deterministic tournament operator analysis, from 500 to 1000 individuals with 100 generations.

Based on the previous experimentation, it was determined that the selection operator for the present investigation is a deterministic tournament, it presents a better behavior, and it requires a smaller number of generations.

As shown in Fig. 1d, the more individuals the population has, the better the evolution, for this reason, it was determined to use 100 generations with 1000 individuals in the experiments.

e. Selection of features with a genetic algorithm and vector support machine as a fitness function in the C10 corpus.

The results obtained from the present investigation are compared with the results of [11] for balanced and imbalanced samples.

As can be seen in Fig. 2, the best results were obtained in two out of three samples, this indicates that the genetic algorithm with a support vector machine as an aptitude function is a good method for the selection of features; in the 5 sample the accuracy is improved by 14.0%, and in the 10 sample the accuracy is improved by 5.3%.

However, in the 50 sample, the proposed method is exceeded by 3.48%. As can be seen in Fig. 2, better results were obtained in one of the samples, in the 10:20 sample, the accuracy is improved by 6.15%.

However, in samples 2:10 and 5:10, the proposed method is exceeded by 7% and 1.7% respectively; the method with imbalanced samples requires that the training sample be larger in order to improve the performance of the classifier.

As can be seen in Fig. 3, the proposed method is compared with other methods used for authorship attribution, such as the proposal in [33] that performs the construction of syntactic graphs, in [29] uses a representation of n-grams of words with Doc2vec, in [34] implements the 15 most referenced works about the authorship attribution task to replicate the methods and analyze whether the methods are replicable.

5 Conclusions

The selection of features does impact the performance of the classifier as shown in Fig. 2a and 2b. In balanced samples, better results are obtained in two of the three samples, improving the accuracy by 5.3% in the 10:10 sample. and 14.0% in the 5:5 sample.

In the imbalanced samples, better results are obtained in one of the three samples, improving the accuracy by 5.65% in the 10:20 sample, when changing the method of selection of features, better results are obtained in three of the six established samples.

The proposed method selects the features by means of a learning method.

According to the experimentation carried out in the present investigation, the following conclusions of the implemented method are determined:

– A genetic algorithm and support vector machine as a fitness function allows selecting a relevant set of features.

– Table two shows the results obtained with the validation documents where it is observed that the method proposed in this research obtains competitive results with the state of the art.

The main contribution of this work is the development of a feature extraction method for the authorship attribution task with a genetic algorithm that implements support vector machine as an aptitude function, this method can be implemented in different language processing tasks natural to reduce dimensionality, improve classification, and identify which are the most significant features.

From the present investigation, an area of opportunity arises with the implementation of a micro heuristic that allows reducing the computational cost of the genetic algorithm as a method for the selection of features.

References

1. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology, vol. 60, no. 3, pp. 538–556 (2009) doi:10.1002/asi.21001
2. Lambers, M., Cor, J.: Forensic authorship attribution using compression distances to prototypes. Computational Forensics, Springer Berlin Heidelberg, pp. 13–24 (2009) doi:10.1007/978-3-642-03521-0_2
3. Pillay, S. R., Thamar, S.: Authorship attribution of web forum posts. ECrime Researchers Summit (2010) doi:10.1109/ecrime.2010.5706693
4. Ramnial, H., Panchoo, S., Pudaruth, S.: Authorship attribution using stylometry and machine learning techniques. Advances in Intelligent Systems and Computing, pp. 113–125 (2015) doi: 10.1007/978-3-319-23036-8_10
5. Zhang, Z., Li, X., Tian, X.: Research on feature weights of Liheci word sense disambiguation. In: Proceedings 8th International Symposium on Computational Intelligence and Design, vol. 2, pp. 7–10 (2015) doi: 10.1109/ISCID.2015.221
6. Shrestha, P., Sierra, S., González, F. A., Montes, M., Rosso, P., Solorio, T.: Convolutional neural networks for authorship attribution of short texts. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, vol. 2, pp. 669–674 (2017)
7. Kern, R., Seifert, C., Zechner, M., Granitzer, M.: Vote/veto meta classifier for authorship identification. In: Proceedings of the Conference on Multilingual and Multimodal Information Access Evaluation (2011)
8. Pillay, S. R., Solorio, T.: Authorship attribution of web forum posts. In: 2010 eCrime Researchers Summit, pp. 1–7 (2010) doi: 10.1109/ecrime.2010.5706693
9. Anwar, W., Bajwa, I., Ramzan, S.: Design and implementation of a machine learning based authorship identification model. Scientific Programming, vol. 2019, pp. 1–14 (2019) doi: 10.1155/2019/9431073
10. De Vel, O., Anderson, A., Corney, M., Mohay, G.: Mining e-mail content for author identification forensics. In: Association for Computing Machinery Special Interest Group on Management of Data, vol. 30, no. 4, pp. 55–64 (2001) doi: 10.1145/604264.604272
11. Plakias, S., Stamatatos, E.: Tensor space models for authorship attribution. In: Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A. (eds) Artificial Intelligence: Theories, Models and Applications. SETN 2008. Lecture Notes in Computer Science, Springer, Heidelberg, vol. 5138, pp. 239–249 (2008)
12. Xue, B., Zhang, M., Browne, W.: A comprehensive comparison on evolutionary feature selection approaches to classification. International Journal of Computational Intelligence and Applications, vol. 14, no. 2 (2015) doi: 10.1142/S 146902681550008X
13. Tan, F., Fu, X., Zhang, Y., Bourgeois, A.: A genetic algorithm-based method for feature subset selection. Soft Computing a Fusion of Foundations, Methodologies and Applications, vol. 12, no. 2, pp.111–120 (2008) doi: 10.1007/s00500-007-0193-8
14. Mohamad, M., Deris, S., Mat, S., Razib, M.: Feature selection method using genetic algorithm for the classification of small and high dimension data. In: Proceeding of the First International Symposium on Information and Communications Technologies, pp. 13–16 (2004)
15. Mikros, G., Perifanos, K.: Authorship identification in large email collections: Experiments using features that belong to different linguistic levels. Notebook for PAN at CLEF (2011)
16. Vilariño, D., Castillo, E., Pinto, D., León, S., Tovar, M.: Baseline Approaches for the Authorship Identification Task. Notebook for PAN at CLEF 201. (2011)
17. Kern, R., Seifert, C., Zechner, M., Granitzer, M.: Vote/veto meta classifier for authorship identification. In: Proceedings of the 2011 Conference on Multilingual and Multimodal

Information CLEF´11: Access Evaluation (Lab and Workshop Notebook Papers), Amsterdam, The Netherlands (2011)
18. Vartapetiance, A., Gillam, L.: Quite simple approaches for authorship attribution, intrinsic plagiarism detection and sexual predator identification. CLEF 2012 Evaluation Labs and Workshop Working Notes Papers (2012)
19. Mesleh, A.: Chi square feature extraction based SVMS Arabic language text categorization system. Journal of Computer Science, vol. 3, no. 6, pp. 430–435 (2007)
20. D'Andrea, A., Ferri, F., Grifoni, P., Guzzo, T.: Approaches, Tools and Applications for Sentiment Analysis Implementation. International Journal of Computer Applications, vol. 125, no. 3 (2015)
21. Varela, P., Martins, A., Aguiar, P., y Figueiredo, M.: An empirical study of feature selection for sentiment analysis. In: 9th conference on telecommunications (2013)
22. Adel, A., Omar, N., Al-Shabi, A.: A comparative study of combined feature selection methods for Arabic text classification. Journal of Computer Science, vol. 10, no. 11 (2014) doi: 10.3844/jcssp.2014.2232.2239
23. Pavlyshenko, B.: Genetic optimization of keywords subset in the classification analysis of texts authorship. In: Journal of Quantitative Linguistics, vol. 21, pp. 341–349 (2014) doi: 10.48550/ARXIV.1211.3402
24. Batista, B., Moreno, J., Moreno, M.: Algoritmos genéticos, una visión práctica, números. Revista de Didáctica de las Matemáticas, vol. 71, pp. 29–47 (2009)
25. Coello, C.: Introducción a los algoritmos genéticos. Soluciones Avanzadas. Tecnologías de Información y Estrategias de Negocios, vol. 3, no. 7, pp. 5–11 (1995)
26. Gestal, M., Rivero, D., Rabuñal, J., Dorado, J., Pazos, A.: Introducción a los algoritmos genéticos y la programación genética (2010)
27. Ponce, P.: Inteligencia artificial con aplicaciones a la ingeniería (2010)
28. Su, S., Li, L., Zhao, Q.: Text feature selection based on improved adaptive GA. In: 7th International Conference on Natural Language Processing and Knowledge Engineering, pp. 169–172 (2011)
29. Posadas-Durán, J., Gómez-Adorno, H., Sidorov, G., Batyrshin, I., Pinto, D., Chanona-Hernández, L.: Application of the distributed document representation in the authorship attribution task for small corpora. Soft Computing, vol. 21, pp. 627–639 (2017)
30. Mirończuk, M., Protasiewicz, J.: A recent overview of the state-of-the-art elements of text classification. Expert Systems with Applications, vol. 106, pp. 36–54 (2018) doi: 10.1016/j.eswa.2018.03.058
31. Ledeneva, Y., García, R.: Generación automática de resúmenes: retos, propuestas y experimentos (2017)
32. Zhang, Z., Li, X., Tian, X.: Research on feature weights of Liheci word sense disambiguation. In: Proceedings of 8th International Symposium on Computational Intelligence and Design, vol. 2, pp. 7–10 (2015) doi: 10.1109/ISCID.2015.221
33. Gómez-Adorno, H., Sidorov, G., Pinto, D., Vilariño, D., Gelbukh, A.: Automatic authorship detection using textual patterns extracted from integrated syntactic graphs. Sensors, vol. 16, no. 9 (2016) doi: 10.3390/s16091374
34. Potthast, M., Braun, S., Buz, T., Duffhauss, F., Friedrich, F., Gülzow, J.M., Köhler, J., Lötzsch, W., Müller, F., Müller, M.E., PaBmann, R., Reinke, B., Retenmeier, L., Rometsch, T., Sommer, T., Träger, M., Wilhem, S., Stein, B., Stamatatos, E., Hagen, M.: Who Wrote the Web? Revisiting Influential Author Identification Research Applicable to Information Retrieval. In: Proceedings of European Conference on Information Retrieval. Lecture Notes in Computer Science, vol. 9626, pp. 393-407 (2016) doi: 10.1007/978-3-319-30671-1_29

ISSN 1870-4069

Improving Leukemia Image Classification by Extracting and Transferring Knowledge by Evolutionary Vision

Rocio Ochoa-Montiel[1,2], Gustavo Olague[3], Humberto Sossa[1,4]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Universidad Autónoma de Tlaxcala,
Facultad de Ciencias Básicas,
Mexico

[3] Centro de Investigación Científica y de Educación Superior de Ensenada,
EvoVision Laboratory Ensenada,
Mexico

[4] Tecnológico de Monterrey,
Escuela de Ingeniería y Ciencias,
Mexico

{ma.rocio.ochoa,gustavo.olague,
humbertososa}@gmail.com

Abstract. The aim of evolutionary vision is to address typical problems of artificial vision through techniques whose principles are based on the theory of biological evolution. This allows us to consider the visual problem as a goal-oriented vision problem. Although the current automatic approaches are widely used in diverse recognition tasks, these are unable to explain how the knowledge to solve the problem is derived. In this paper, we use an evolutionary vision technique called brain programming (BP) to extract the knowledge used for the classification task. This model allows us to know how the knowledge to solve the leukemia images classification problem is derived. In addition, we present two variants focused on transferring knowledge to improve the classification. Results show that classification on leukemia images is achieved successfully from the solutions obtained by the proposed variants.

Keywords: Evolutionary vision, leukemia type classification, knowledge transfer.

1 Introduction

Leukemia is a cancer of the early blood-forming cells that is among the top 10 types of cancer more frequent in the world. Its diagnosis requires of information derived from several modalities, including morphology, cell phenotyping, cytochemistry, cytogenetics, and molecular genetics. Particularly, the morphological analysis requires a manual microscopic examination that is time-consuming and is prone to human error [4, 10].

To overcome the limitations of the manual analysis, several computational methods have been proposed following both, handcraft and automatic approaches. A handcraft approach utilizes conventional image processing and machine learning techniques, in which properties as the shape, color, and the distribution of some elements in the image are meaningful details for the recognition task [1, 2]. The knowledge derived from this approach can be used for the development of explainable learning models.

On the other hand, although the automatic approaches such as the convolutional neural networks have shown satisfactory results in leukemia images recognition, these are not driven by human reasoning, and they are computationally expensive, which is a clear drawback [7]. In this regard, the evolutionary computing techniques as the genetic programming offer an important advantage because use a trees-based representation that can automatically perform feature extraction, feature derivation, feature selection, and classification. Nevertheless, the classification task is a challenging problem due to issues such as low classification accuracy and a long training time.

For this reason, the reuse of learned knowledge is helpful to address these limitations as is suggested in [3, 6]. In this paper, we use an evolutionary insight paradigm as a baseline method to address the leukemia classification problem and show that knowledge extraction and transfer are useful for improving classification accuracy even when the reused knowledge comes from classes other than those under study. In the next section, we recall the theoretical concepts. Section 3 presents the proposed methodology. In Section 4, experiments and results are shown. Conclusions are drawn in Section 5.

2 Materials and Methods

In this work, the recognition problem is addressed using a model of evolutionary computer vision named brain programming (BP) as the baseline method. The BP uses the genetic programming (GP) to discover a set of evolutionary visual operators (EVOs) that are included within a structure named artificial visual cortex (AVC) [9].

We introduce the image recognition problem from the standpoint of data modelling. A minimization problem requires finding a solution $\mathbf{L}_{min} \in S$ such that $f(\mathbf{L}_{min})$ is a global minimum on S, $\exists \mathbf{L}_{min} \in S : f(\mathbf{L}_{min}) \leq f(\mathbf{L})$. In GP for a recognition problem, the purpose is to find a function that satisfies the task of data modelling. Thus, image recognition is defined as:

$$\mathbf{y} = \min(f(\mathbf{x}, \mathbf{F}, \mathbf{T}, \mathbf{a})), \tag{1}$$

where the dataset is given by $(\mathbf{x}, \mathbf{y})$, $\mathbf{F}$ denotes the set of functions, $\mathbf{T}$ represents the terminal set, and $\mathbf{a}$ describes the parameters used for tuning the algorithm. Thus, we require a feature extraction method and a suitable criterion $\mathbf{C}$ for minimization. The methodology requires the definition of two parts: 1) the AVC is the algorithm in charge of feature extraction, and 2) BP is the algorithm used to tune $(\mathbf{F}, \mathbf{T}, \mathbf{a})$ for each visual operator into the AVC. We use a multilayer perceptron (MLP) as the classifier to learn a mapping $f(\mathbf{x})$ where the descriptors $\mathbf{x}_i$ are associated with labels $\mathbf{y}_i$. In this work, we address the given problem as a multiclass classification task.

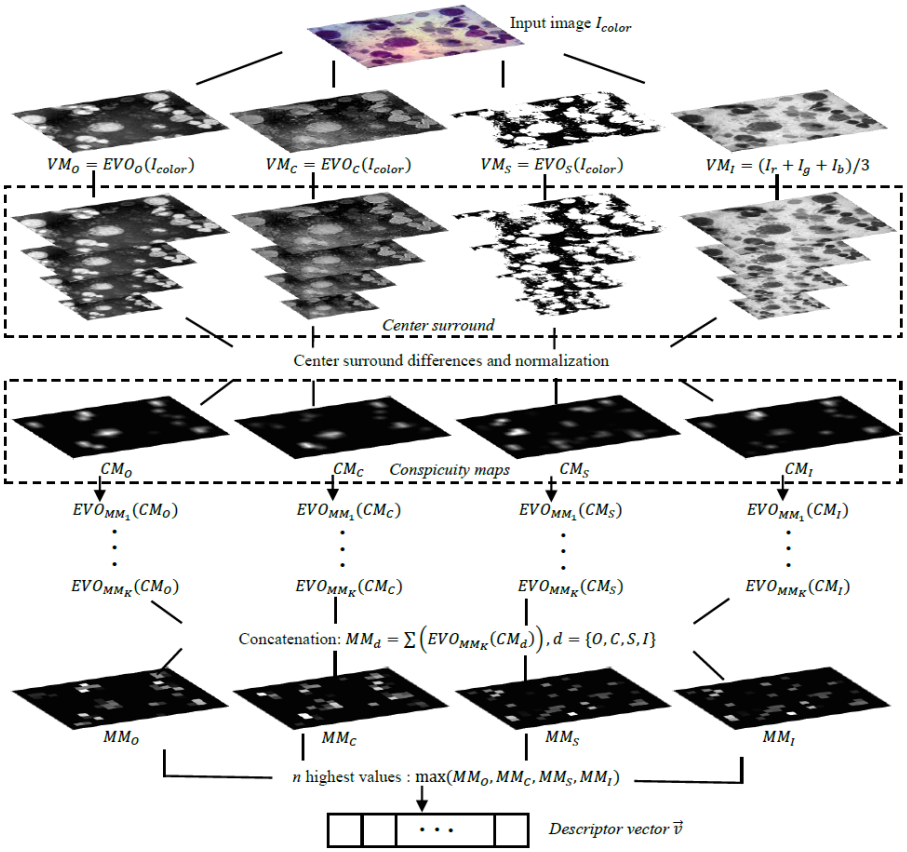


Fig. 1. Flowchart of the first stage of AVC.

Hence, it is assumed that in the minimization problem, the variables $((\mathbf{x}, \mathbf{y}), \mathbf{F}, \mathbf{T}, \mathbf{a}, \mathbf{C})$ are related in such a way that the objective is to associate the descriptors and the labels. Since the function is the base for understanding the transformations of the scene, then an image I is considered as the graph of a function [9].

2.1 Artificial Visual Cortex

The AVC is a hierarchical model that reproduces some aspects of the human visual cortex. First, a set of visual functions is defined according to the problem, and then each layer of the artificial visual cortex computes the mathematical operations pertaining to these functions. A representation of the object of interest is created from the image's visual features. This representation consists of salient points in the image to generate a descriptor used for classification. Figure 1 shows the first stage performed by the AVC. The AVC is composed of two stages. In the first stage, the features that describe

the object are acquired and transformed, whereas in the second phase, the descriptor obtained in the previous stage is used to classify the object.

The input of the AVC is an RGB image I, and multiple colour channels are considered to create the set $I_{color} = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$, where the elements refer to the colour components of the RGB, HSV, and CMYK colour spaces. Later, each EVO is applied separately to generate a visual map (VM) to emphasize features based on orientation, color, and shape.

The EVOs are specialized functions evolved from a set of image operators such as color opponency or Gabor filters that, according to the knowledge from neuroscience-based on the way these functions are obtained in the biological visual cortex of the brain. In this way, the evolutionary process uses a set of functions and terminals to generate the operators for each dimension.

The dimensions d are the feature types used by the AVC, in this work we use the dimensions of orientation, color, shape, and intensity. This last, is the only one that performs a constant operator due to it only transforms the input color image into an intensity image. Thus, $VM_I = (I_r + I_g + I_b)/3$. Where I_r, I_g, and I_b are the color components of the RGB color model. The next step is to compute a centre surrounding process. First, the scale-invariant features are extracted and stored in a conspicuity map (CM).

The CM is calculated as the difference between the different scales that are obtained through a pyramid of 9 levels: $P_d^\sigma = \{P_d^{\sigma=0}, P_d^{\sigma=1}, P_d^{\sigma=2}, ..., P_d^{\sigma=8}\}$. A Gaussian smoothing filter is used on each VM to calculate each pyramid. This produces an image that is half the size of the input map. The process is repeated 8 times to obtain the 9-level pyramid. In the next step, the differences with respect to each pyramid level in P_d^σ are calculated using (2) as follows:

$$Q_d^j = P_d^{\sigma=\left\lfloor \frac{j+9}{2} \right\rfloor+1} - P_d^{\sigma=\left\lfloor \frac{j+2}{2} \right\rfloor+1}, \tag{2}$$

where $j = 1, 2..6$. Each level of P_d^σ is normalized and scaled to the dimensionality of the VM using polynomial interpolation. Finally, the six levels are combined into a single map with a summation operation. As a result, a CM is obtained for each dimension. The second stage of the AVC is description and classification. Here, a map is built to highlighting the most salient features from the CMs. For this, a mental map (MM) is built from the CMs using (3), where d is the dimensionality and k is the cardinality of the set EVO_{MM}. The MMs occupy the fourth position of the tree onward:

$$MM_d = \sum_{i=1}^{k} EVO_{MM_i}(CM_d). \tag{3}$$

Once the MMs are obtained and concatenated with the remainder of the syntactic trees, the generated program is applied to each image. The n highest values are used to define the descriptor vector $\overrightarrow{v}$ for each image. The next step is to train a classifier using the feature vectors. In this work, an MLP is trained to create a model $f(\mathbf{x})$ that maps a set of descriptor vectors $\mathbf{x}_i$ to their corresponding labels $\mathbf{y}_i$, satisfying (1).

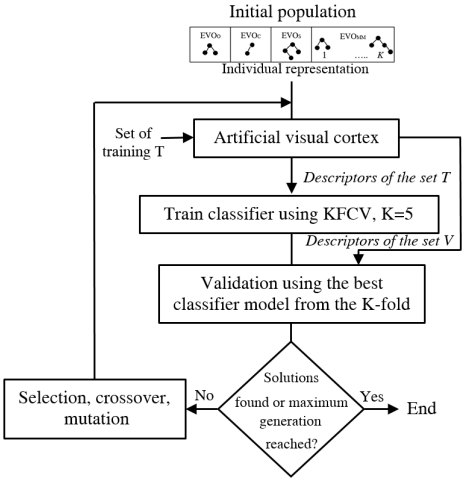


Fig. 2. Baseline method.

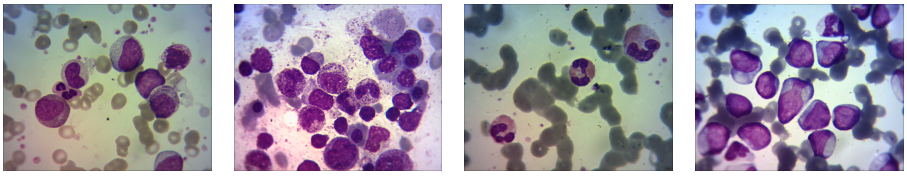
3 The Proposed Methodology

The Brain programming is the baseline method that we use to search for a set of mathematical expressions inspired on the functionality of specialized tissues in the brain in order to solve the classification problem. The evolutionary process of BP begins with a randomized initial generation. In this way, a set of initialization variables are defined, such as the population size, sizes of solutions or individuals, and crossing-mutation probabilities.

An individual represents a computer program written with a set of syntactic trees included in hierarchical structures. These individuals contain four kinds of functions that are defined in previous research by[9], one for each visual operator (VO). Each individual is encoded in a multitree representation. A variable number of syntactic trees, ranging from four to 10, compose each individual. The individuals are evaluated for each AVC performed, and a classifier determines their fitness for each generation.

Selection, crossover, and mutation processes are performed according the individual representation as suggested in [9], thus we consider that the whole individual is a chromosome, and each EVO within this is a gene. Finally, the stopping conditions are: (a) the algorithm reaches a maximum number of generations, or (b) the algorithm fitness reaches an optimal value; this means, all images are correctly classified. Fig. 2 presents the general scheme of the proposed approach.

Although in evolutionary computation it is common the use random sources to start the evolution, we propose to adapt the baseline method described above, in a similar way as in [8] to discover the best solutions for the classification task. To achieve this, we built a testbed with three experiments. In all experiments, we use the baseline method to classify the leukemia images M2, M3, M4, and M5. The first experiment uses only the line base method, whereas the second experiment uses the best solutions from the first experiment as the initial population.



(a) M2 (b) M3 (c) M4 (d) M5

Fig. 3. Types of acute myelocytic leukemia.

For the third experiment, in the first step, we use the baseline method to classify three different types of leukemia L1, L2, and L3. From the best solutions of this step, is created the initial population for the third experiment.

4 Experiments and Results

The implementation is carried out on a computer with an Intel i9-7900X, 64 GB of RAM, the Windows10 operating system, and MATLAB. Parameters initialization is as follow: population size 30, generations 30, method ramped half and half for generating of population, crossover rate 0.4, roulette wheel selection, mutation 0.1, tree max depth 50 levels, and a maximum length of genes 10. From the diagram in Figure 2, the evolutionary loop starts by computing the fitness of each AVC obtained from each individual using an MLP to calculate the classification rate using the training T and validation V sets.

The MLP has one intermediate layer with 50 neurons. Later, a set of AVCs is selected using roulette-wheel selection, and the best AVC is used for further processing. The new individual is created from the selected AVC by applying a crossover or mutation operation, as in [9]. The conditions to stop are (1) when the classification rate is equal to 100% or when (2) the number of generations reaches $N = 30$.

The dataset used is composed of 868 bone marrow smear color images from four subtypes of acute myeloid leukemia (AML): M2, M3, M4, and M5. We use 217 images per class, Fig. 3 shows a sample for each class. Image acquisition is performed by employing an optical microscope with a magnification of 1250 times and a camera coupled to the microscope. The images size is 1280×1024 pixels, which are resized to 256×320 pixels using bicubic interpolation.

For the experiments, we divide the dataset into three parts: a training set T, a validation set V, and testing set A. Each new individual is estimated by the classification error rate with the MLP from the best classifier model B of the k-folds. To select the best-performing solution, we test B on the validation set V. Later, we select one solution with the best validation error as the (near-) optimal feature descriptor, which is used for the model evaluation.

4.1 Results

From the proposed methodology, in order to assessment the model we test the best solution on the test A set using five-fold cross-validation. The average of the five test accuracies is the classification result.

Table 1. Results of classification.

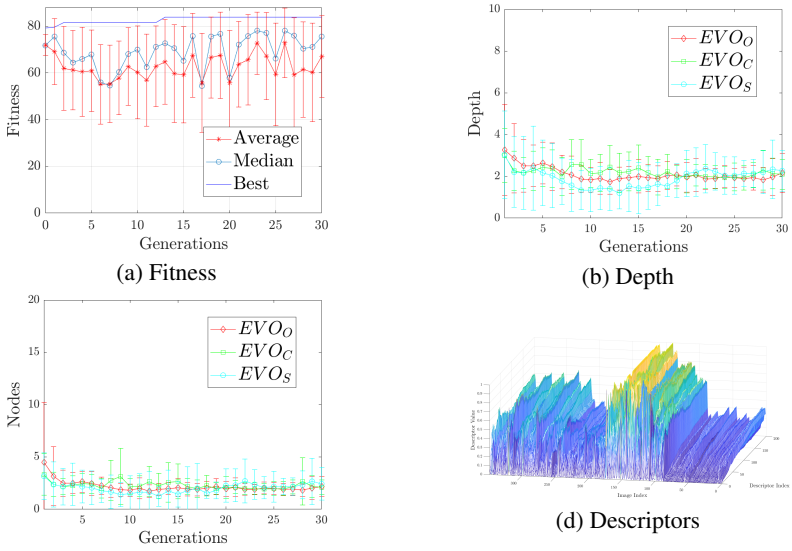
	Standard evolution of four classes E1			**Evolution from** four known classes E2			**Evolution from** three unknown classes E3		
Run	Val. accuracy	Test accuracy		Val. accuracy	Test accuracy		Val. accuracy	Test accuracy	
		MLP	Random forest		MLP	Random forest		MLP	Random forest
1	77.31	72.67	75.82	81.92	76.34	79.49	78.84	79.38	**82.04**
2	73.46	66.07	69.71	83.84	80.31	**81.16**	77.69	70.05	73.76
3	72.69	72.24	74.15	84.61	79.83	80.71	76.92	74.01	78.08
4	81.92	77.36	**79.96**	81.53	77.91	79.99	78.84	70.50	69.32
5	80.00	73.25	76.53	83.07	72.18	75.76	77.69	76.77	80.35
6	71.92	68.90	72.14	81.92	76.92	79.83	-	-	-
7	78.07	75.35	76.93	81.92	78.15	79.68	-	-	-
8	75.38	72.08	74.22	81.92	75.36	79.89	-	-	-
Average	76.34	72.23	74.93	82.59	77.13	**79.57**	77.99	74.14	76.71
σ	± 3.59	± 3.51	± 3.13	± 1.12	± 2.60	± 1.63	± 0.83	± 4.01	± 5.16
Outliers detected	0	0	0	0	0	1	0	0	0
Critical value Z	2.12	2.12	2.12	2.12	2.12	2.12	1.71	1.71	1.71

We repeat this process ten times and, the overall classification result is the accuracies' average of these repetitions. To make a more robust evaluation, we performed the independent test using two classifiers: an MLP and a random forest. Table 1 shows the results of the model evaluation. Seeing the results, we confirm the advantage of applying the best solutions discovered in previous experiments, despite using solutions from unknown leukemia types for the classification problem.

We consider the Grubbs statistical method for outlier detection [5].We run the experiments using Grubbs' test from the results of every experiment, with a significance level of 0.05 (two-sided). Values of average, standard deviation, outliers detected, and critical value Z are reported in Table 1. According to the results, there are no outliers, except in the experiment E2. Nevertheless, this is not significant because only one outlier was detected. In view of the best solutions are obtained by using the best individuals as the initial population using only the baseline method, we selected the best solution from run two to show the behavior during the evolutionary process.

The validation accuracy of this solution is 83.84%, and the test accuracy is 81.16% using random forest. This is a competitive result since the content of the images is visually similar. The fitness evolution, the average, standard deviation, and the best fitness thus far along the run are shown in Fig. 4 (a). Furthermore, in order to quantify the solution complexity we use the depth and the number of nodes. In this way, the Fig. 4 (b)-(c) depicts that the discovered solution has low complexity because the number of nodes and depth of trees decreases as the generations progress.

On the other hand, the properties of descriptor generated by the best solution in Fig. 4 (d) illustrate the range of descriptor values for this solution. The descriptor vector given by the AVC is a numerical vector of length 200. Since we use a balanced data set with the following percentages for each set: 30% for set T, 30% for set V, and 40% for set A; thus, set A contains 348 images.



(a) Fitness (b) Depth (c) Nodes (d) Descriptors

Fig. 4. Behavior of the best solution. (a) Fitness, (b-c) Complexity, (d) Descriptors.

Table 2. Structure of best solutions.

No. Exp.	EVO_o	EVO_c	EVO_s	EVO_{MM_d}
$E1$	$log(K)$	S^2	S	$MM_1 = \sqrt{D_y(CM_d)}$
				$MM_2 = D_x(D_x(D_x(D_x(D_x(D_x(CM_d))))))$
				$MM_3 = CM_d/(D_x(D_x(CM_d)))$
$E2$	$log(K)$	S^2	$round(S)$	$MM_1 = D_y(CM_d)$
				$MM_2 = D_x(D_x(D_x(D_x(D_x(D_x(CM_d))))))$
				$MM_3 = CM_d/(D_x(D_x(CM_d)))$
$E3$	$G_{\sigma=1}(S)$	$Op_{b-y}(I_{rgb}) * Op_{b-y}(I_{rgb})$	B	$MM_1 = Half(G_{\sigma=1}(G_{\sigma=1}(CM_d)))$
				$MM_2 = G_{\sigma=1}(CM_d)$
				$MM_3 = \frac{\sqrt{G_{\sigma=1}(G_{\sigma=1}(CM_d))} * G_{\sigma=1}(CM_d)}{G_{\sigma=1}(G_{\sigma=1}(CM_d))}$

Then, in Fig. 4 (d) the indices of images 1...87 correspond to class M2, and so on for classes M3, M4, and M5. Clearly, the descriptor values show the differences between the categories, which in some cases could be evaluated with simple techniques instead of using a classifier. Finally, in Table 2 we show the structure of the best solutions for each experiment. From these results, it is worth to note that the solutions can be read and are susceptible to simplifying.

In addition, as expected, experiments E1 and E2 give similar solutions due to the similarity of the images used in both cases. From results, we can see that the use of the best solutions from previous experiments like in E2 and E3, is useful to improve the classification results. Furthermore, the model used as baseline technique allows us to know how knowledge is derived to obtain the best solution.

This is a significant advantage, since experts need to recognize leukemia types from the characteristics of an image, as well as to know the process for learning to recognize them. Although we do not find a similar approach to our proposal, we provide experiments with a convolutional neural network for comparison. The network structure consists of four convolutional and max-pooling layers with 4, 8, 16, and 32 filters, respectively. Data augmentation is used on the training set, which includes horizontal and vertical reflexion.

Five-fold cross-validation is used to assess the training performance, which is stopped when the validation loss does not decrease for 20 epochs. Later, the net best model from the five-fold validation is selected and used with the test set. To evaluate the global classification performance, we repeat the experiment eight times, from which accuracy of $86.85\% \pm 5.88$ (mean $\pm$ s.d., runs=8) is obtained. From the results, it is worth noting that although the CNN achieves higher performance on the classification task, a critical problem is the lack of information about the learning process.

5 Conclusions

This paper presents an evolutionary vision model named brain programing, as a baseline method to solve the problem of leukemia recognition. An important advantage of this model is its capability for explaining how the knowledge to solve the problem is derived, which opens the possibility of answer the question: What is the visual task for?

Since a hierarchical structure and a functional composition perform the visual extraction and description, then the evolutionary cycle discovers the functions embedded in this structure. In two of the experiments shown, we use the best set of solutions from previous experiments as the initial population for a new set of experiments. The results showed that knowledge transfer through these good solutions improves the classification rate despite the fact that the solutions used as the initial population correspond to different types of leukemia from those to be classified.

Thus, it is to note that the use of knowledge transfer can improve the classification performance on the complex image classification task, in both similar domains and different domains. Due to the recognition problem requires a clear explanation to better understand the studied subject, the presented model can be helpful in the medical area for the recognition of diverse pathologies. Finally, future work from these results is to widen the number of different domains for knowledge transfer and address the issue of explainability in this knowledge transfer.

Acknowledgments. The authors would like to acknowledge the support provided by the Instituto Politécnico Nacional under projects 20200630 and 20210788, CONACYT under projects 65 (Fronteras de la Ciencia) and 6005 (FORDECYT-PRONACES), and CICESE through project 634-135 to carry out this research. The first author thanks the Autonomous University of Tlaxcala, Mexico for their support. The authors also express their gratitude to the Applied Computational Intelligence Network (RedICA).

References

1. Aljaboriy, S., Sjarif, N., Chuprat, S., Abduallah, W.: Acute lymphoblastic leukemia segmentation using local pixel information. Pattern Recognition Letters, vol. 125 (2019) doi: 10.1016/j.patrec.2019.03.024
2. Bodzas, A., Kodytek, P., Zidek, J.: Automated detection of acute lymphoblastic leukemia from microscopic images based on human visual perception. Frontiers in Bioengineering and Biotechnology, vol. 8, pp. 1005 (2020) doi: 10.3389/fbioe.2020.01005
3. Dinh, H., Chu, T., Nguyen, Q. U.: Transfer learning in genetic programming. pp. 1145–1151 (2015) doi: 10.1109/CEC.2015.7257018
4. Ferlay, J., Ervik, M., Lam, F., Colombet, M., Mery, L., Piñeros, M., Znaor, A., Soerjomataram, I., Bray, F.: Global cancer observatory: Cancer today. lyon, france: International agency for research on cancer. https://gco.iarc.fr/today
5. Grubbs, F. E.: Sample criteria for testing outlying observations. Ann. Math. Statist., vol. 21, no. 1, pp. 27–58 (1950) doi: 10.1214/aoms/1177729885
6. Iqbal, M., Xue, B., Al-Sahaf, H., Zhang, M.: Cross-domain reuse of extracted knowledge in genetic programming for image classification. IEEE Transactions on Evolutionary Computation, vol. 21, no. 4, pp. 569–587 (2017) doi: 10.1109/TEVC.2017.2657556
7. Ochoa-Montiel, R., Olague, G., Sossa, H.: Expert knowledge for the recognition of leukemic cells. Applied Optics, vol. 59, no. 14, pp. 4448–4460 (2020) doi: 10.1364/AO.385208
8. Olague, G., Chan Ley, M.: Hands-on artificial evolution through brain programming, pp. 227–253 (2020)
9. Olague, G., Clemente, E., Dozal, L., Hernández, D.: Evolving an artificial visual cortex for object recognition with brain programming. In: Schuetze, O., Coello, C. A., Tantar, A. A., Tantar, E., Bouvry, P., Moral, P. D., Legrand, P. (eds) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III, Studies in Computational Intelligence, vol. 500, pp. 97–119 (2014)
10. Rodak, B. F., Carr, J. H.: Clinical Hematology Atlas (2016)

ISSN 1870-4069

Classical Contrast Enhancement Methods in the Classification of Estrous Cycle Images

Rocio Ochoa-Montiel[1,2], Ismael Llamur[3],
Humberto Sossa[1,4], Gustavo Olague[5]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Universidad Autónoma de Tlaxcala,
Facultad de Ciencias Básicas,
Mexico

[3] Universidad Nacional de Tucuman,
Facultad de Ciencias Exactas y Tecnologías,
Argentina

[4] Tecnológico de Monterrey,
Escuela de Ingeniería y Ciencias,
Mexico

[5] Centro de Investigación Científica y de Educación Superior de Ensenada,
EvoVision Laboratory Ensenada,
Mexico

{ma.rocio.ochoa, ismaelllamur013, humbertososssa,
gustavo.olague}@gmail.com

Abstract. In the biological area, the short reproductive cycle in rodents is useful because it allows analyzing electrophysiological properties, behaviors, or drugs effects, through the changes observed during this period. This cycle is composed of 4 stages, in which the classification is determined by vaginal cytology. Although automatic approaches have been used for the recognition of these stages, they are computationally expensive and require a great number of images for adequate performance. In this work, we study the effect of contrast enhancement on the images classification of the reproductive cycle named estrous cycle. We use a dataset of 344 images and four classical contrast enhancement methods. We extract texture features and use four classifiers to evaluate the impact of the contrast enhancement methods. From the results, we find that the contrast enhancement methods that do not emphasize strongly some regions in the images show higher classification results than those yes do it. Furthermore, features extracted manually overcome the classification rate concerning the features extracted automatically with a standard convolutional neural network.

Keywords: Contrast enhancement, classification, estrous cycle.

1 Introduction

Artificial vision has a wide variety of applications in diverse fields like surveillance, security or medicine [5, 7]. In Biology, the analysis of rodent tissues is especially useful to study physiological processes due to these are carrying on short periods. Particularly, reproduction is an ideal process for researching changes along a cycle, such as fertility rates, effects of treatments, or environmental diverse conditions, among others [13, 10]. The reproductive cycle in rodents named the estrous cycle is formed by four phases: proestrus P, estrus E, metestrus M, and diestrus D.

Knowledge of these stages is important for interpretations of female animals' data, whereas their identification is through the observation of cells in vaginal smears where properties like type, number, shape, size, and proportion of cells are evaluated [2, 4]. Nevertheless, the recognition of estrous cycle stages by a human expert takes a long time, and the evaluations are frequently subjective due to differences of skill of the examiners. We assume this is one of the main reasons why the automatic recognition problem has been poorly addressed, since the amount of correctly labeled data is critical for the good performance of these approaches.

In this regard, [9] presents a quantitative method for assessing Estrous cycle stages. However, the method focuses on showing trends between diverse cell types in each stage. On the other hand, authors in [12] propose software that enables more efficient cycle stage data analysis and pattern visualization, while in [8] the visual classification of the estrous cycle images is addressed by using support vector machines, multilayer perceptron networks, and convolutional neural networks.

On the other hand, [14] proposes the deep learning-based classification of the estrous cycle, considering only three stages of the cycle. In this work, for the first time we evaluate some classical contrast enhancement methods on the classification of estrous cycle images. It is important to mention that our objective is to know the impact of these methods, and not to exceed the classification rate of previous works [8, 14]. In the next section, the materials and methods are described. Section 3 presents the proposed methodology. In section 4, experiments and results are shown. Conclusions are included in section 5.

2 Materials and Methods

2.1 Contrast Enhancement Methods

The simplest kinds of image processing are point methods, where each output pixel value depends on an input pixel value; including some globally recovered information or parameters. Common techniques of image enhancement as histogram equalization (HE), adaptive histogram equalization (AHE), and local saturation (LS) are in this category.

We decided to use these because the aim is to enhance details over small or regular areas in the images [15]. HE enhances the contrast of images by transforming the values in an intensity image so that the histogram of the output image approximately matches a specified histogram, uniform distribution by default. Whereas, AHE performs contrast-limited adaptive histogram equalization.

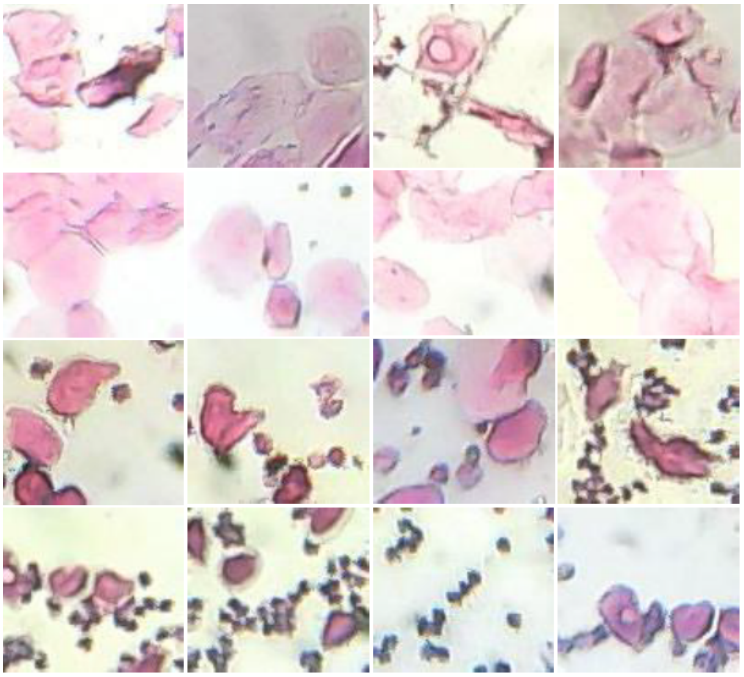


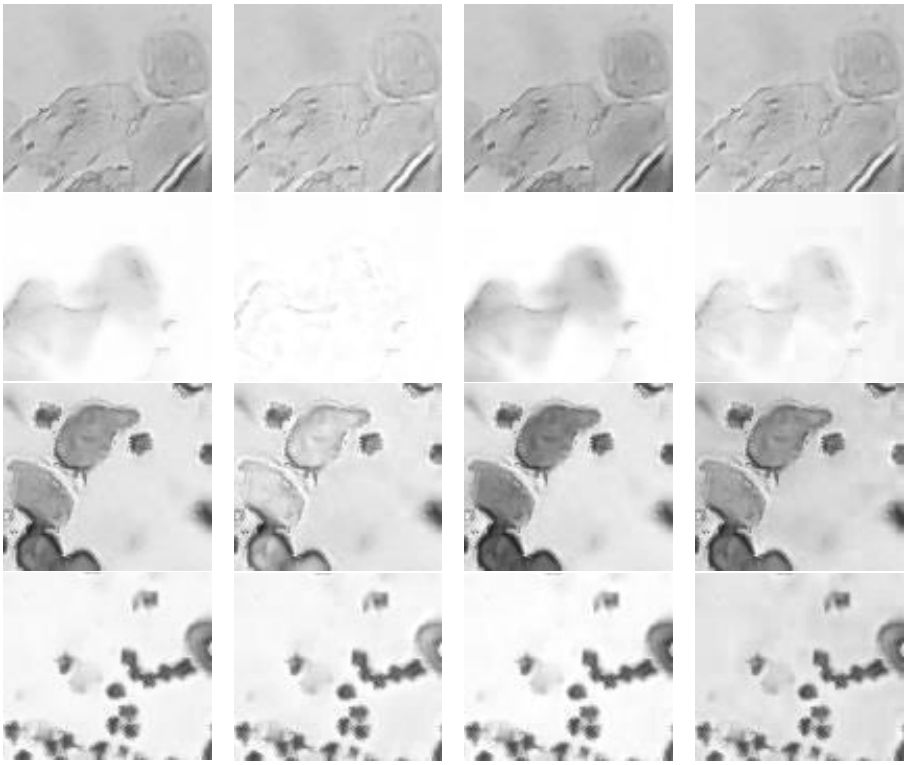
Fig. 1. Representative cells of the stages of reproductive cycle in rats. Images in each row depict the each cycle stage: row 1-stage P, row 2-stage E, row 3-stage M and row 4-stage D.

Unlike HE, it operates on small data regions (tiles) rather than the entire image. Each tile's contrast is enhanced so that the histogram of each output region approximately matches the specified histogram, typically a uniform distribution. The contrast enhancement can be limited to avoid amplifying the noise which might be present in the image. On the other hand, LS increases the image contrast by mapping the values of the input intensity image to new values, hence $n\%$ of the data is saturated at low and high intensities of the input data.

Furthermore, we use an automatic contrast enhancement (ACE) proposed in previous work [11], which enhances the contrast on local areas through a histogram approximation using differential evolution. An advantage of this method is the possibility of defining in which areas of the image we want to contrast enhancement. Thus, clearer or darker regions (Cr or Dr) can be selected for the enhancement.

2.2 Classifiers

The most of classifiers used in machine learning require features extracted previously in a manual or semi-automatic way. Classifiers as multilayer perceptron (MLP), random forest (RF), and the smooth dendrite morphological neurons (SDMN) are of this type. The firsts are known for their good performance in diverse applications, whereas the SDMN is a last generation net recently proposed by [6] that has a good generalization capacity.



(a) Grayscale (b) Band R (c) Band G (d) Band B

Fig. 2. Images of bands R, G, B. Each row depict a cycle stage: row 1-stage P, row 2-stage E, row 3-stage M and row 4-stage D.

On the other hand, the convolutional neural networks (CNNs) are the approaches most representative of automatic classification due to they automatically extract features and perform classification. These models do not need previous image processing or explicit feature extraction. Instead, they use the image to find adequate descriptors for image classification. Thus, it is assumed that the learning occurs by using the position of patterns directly from the input image data exclusively, without considering previous knowledge about this.

3 The Proposed Methodology

In this work we use a balanced dataset of 344 images with 86 for each stage of the estrous cycle. Images were acquired with an optical microscope using a magnification of 400X and a camera Logitech C170. These images are in .jpg format with a resolution of 100×90. Fig. 1 presents images RGB for each stage. From the dataset, we use the G band images because this shows better contrast concerning the R and B bands, as is shown in Fig. 2.

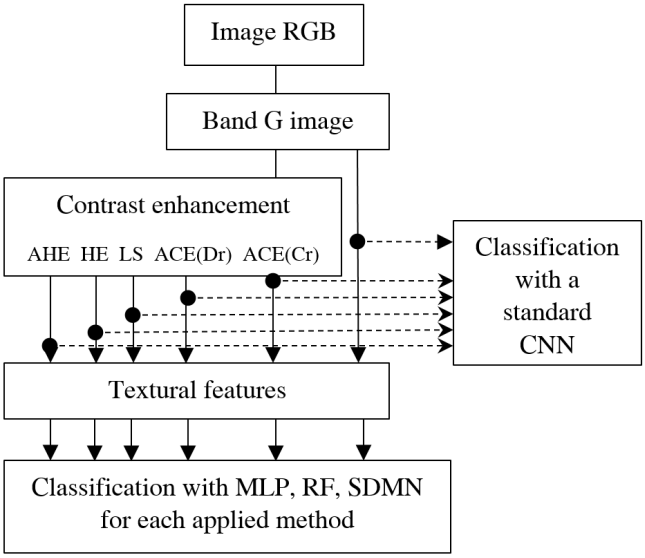


Fig. 3. General flowchart of the proposed methodology.

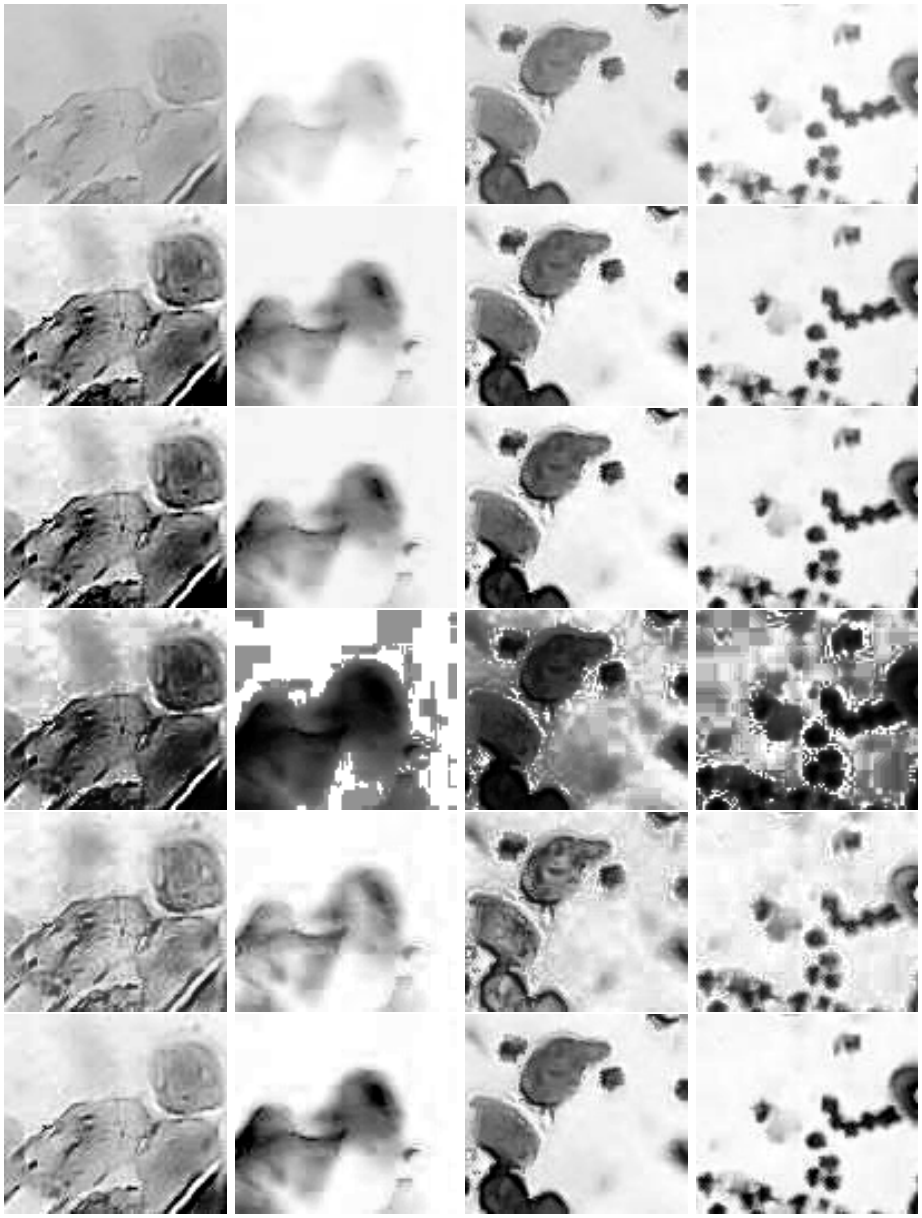
First, we apply the contrast enhancement methods mentioned in the previous section. In addition, we consider images without applying the contrast enhancement to compare. After that, textural features based on the gray level co-occurrence matrix (GLCM) are extracted. These features are used by the classifiers MLP, RF, and SDMN to know the advantages of the contrast enhancement methods in image classification. On the other hand, to compare with an automatic classification method, we use a standard CNN. We assessment the classification performance for each image class P, E, M, and D, which depict the four phases of the estrous cycle. In this way, to test the classifiers MLP, RF, and SDMN we use textural features since the images are in grayscale; whereas CNN uses the images directly. Fig. 3 shows the general flowchart of the proposed methodology.

4 Experiments and Results

The experiments were performed on a CPU Intel Core i9- 7900X, 64GB RAM, Windows10 Enterprise Edition operating system, graphics processing unit (GPU) GeForceGTX 1080, and MATLAB.

4.1 Datasets

Datasets used in the experiments are built applying the contrast enhancement methods ACE (Cr, and Dr), HE, AHE, and LS from section 2. ACE was implemented according to [11], and the methods remainder are computed with the functions *histeq*, *adapthisteq*, and *imadjust*, respectivetly from the image processing toolbox of Matlab.



(a) Stage D (b) Stage E (c) Stage M (d) Stage D

Fig. 4. Images of datasets used. Row 1. Band G, row 2. ACE (Cr), row 3. ACE (Dr), row 4. HE, row 5. AHE, and row 6. LS.

Additionally, a dataset is built from the images of band G without applying contrast enhancement. In this way, we obtain six datasets for the experiments. Fig. 4 shows the images obtained from the methods applied.

Table 1. Results of classification.

	SDMN		**MLP**		**RF**		**CNN**	
Method	Accuracy (%)	s.d	Accuracy (%)	s.d	Accuracy (%)	s.d	Accuracy (%)	s.d
Band G	62.50	± 3.55	69.23	± 4.04	64.42	± 2.82	**68.27**	± 4.26
AHE	68.33	± 4.88	**70.19**	± 3.91	61.54	± 1.78	62.50	± 7.74
HE	61.67	± 3.01	53.85	± 3.04	55.77	± 2.43	49.04	± 9.27
LS	69.17	± 5.18	67.31	± 2.80	**69.23**	± 3.43	58.65	± 5.04
ACE (Dr)	69.17	± 4.90	63.46	± 1.50	61.54	± 2.87	48.08	± 5.64
ACE (Cr)	**71.25**	± 4.49	67.31	± 3.71	61.54	± 2.02	48.08	± 8.19
Average	67.02	± 4.34	65.23	± 3.17	62.34	± 2.56	55.77	± 6.69
Min	61.67	± 3.01	53.85	± 1.5	55.77	± 1.78	48.08	± 4.26
Max	71.25	± 5.18	70.19	± 4.04	69.23	± 3.43	68.27	± 9.27

4.2 Feature Extraction and Classification

The textural features for the tests with MLP, RF, and SDMN are computed from the gray level-co-occurrence using eight gray levels. Because the GLCM probabilities represent the conditional joint probabilities of all pairwise combinations of gray levels in the spatial windows of interest given the parameters interpixel distance (δ) an orientation (θ), we adjust $\delta = 1$, and $\theta = \{0, 45, 90, 135, 180\}$. We use seven shift-invariant features suggested by [3]: uniformity, entropy, dissimilarity, inverse difference, inverse difference moment, and correlation. These features are computed for each value of θ.

Later, we average the four interpixel orientations for each feature, thus we obtain seven features for each image. These features are used independently by each classifier. For the MLP, we consider one hidden layer with 50 neurons and the activation function traingdx from the Matlab toolbox. RF was initialized with 500 trees and the classification is performed by majority vote as in [1]. Finally, SDMN was used with the parameters suggested in [6]. For the experiments with the CNN, we use a structure composed by four convolutional and max-pooling layers.

The feature maps are flattened and reduced to an output of size four. Data augmentation is used on the training set. The augmentation operations include horizontal and vertical reflexion. Accordingly, we have the addition of a random number of augmented images to the training set in each epoch. The number of filters in the four convolutional layers are 4, 8, 16, and 32. The training process is stopped when the validation loss does not decrease for 20 epochs. In all experiments, we use 50% of data for training, 20% for validation, and the remainder for testing.

Five-fold cross-validation is used to assess the training performance, and the best net model from the five-fold validation is selected and used with the test set to assess the net performance. To evaluate the global classification performance, we repeat the experiment ten times. Results of classification experiments for the datasets—Band G, AHE, HE, LS, ACE(Dr), ACE(Cr)— are presented in Table 1. Note that results in ACE(Cr) reached the best performance.

				%Acc./class
45	1	8	6	94.11
0	55	0	5	66.66
12	1	38	9	72.22
8	10	9	33	76.47

Fig. 5. Confusion matrix of the best solution. Row1: stage P, Row2: stage E, Row3: stage M, Row4: stage D.

In general the methods in which the output image does not tend to binarization are adequate for classifying when texturales features are used. On the other hand, the average by classifier shows that classifiers with better performace are SDMN and MLP, whereas the worst performance is achieved by the CNN.

Furthermore, we show in Fig. 5 the confusion matrix for the best solution (this is, accuracy=71.25 for SDMN in Table 1). We observed that the stage P is the best recognized, while the stage E obtains poor classification results. Regard CNN, the features are automatically extracted by the net, and the contrast enhancement methods are not useful for classifying, as we can see Table 1. Finally, it is to note that method HE shows poor performance in all tests.

5 Conclusions

This work proposes for the first time the use of contrast enhancement methods for classifying the four stages in the reproductive cycle on rodents. The methodology proposed is inspired by previous works, in which the main objective is to improve the classification rate. Although this aim is important, in this work we focus on knowing the impact of classical contrast enhancement methods for the estrous cycle image classification.

From the results, we suggest that using contrast enhancement methods is useful when features are extracted manually to be used by classifiers. While for the automatic classifiers these methods are not useful. In a future work, we like to improve the classification rate, and to evaluate another automatic classification methods.

Acknowledgments. Authors would like to acknowledge the support provided by the Instituto Politécnico Nacional under projects: SIP 20200630 and SIP 20210788, CONACYT under projects: 65 (Fronteras de la Ciencia) and 6005 (FORDECYT-PRONACES), and CICESE through the project 634-135 to carry out this research. First author thanks the Autonomous University of Tlaxcala, Mexico for the support. Authors also express their gratitude to the Applied Computational Intelligence Network (RedICA).

References

1. Breiman, L.: Random forests. Machine Learning, vol. 45, no. 1, pp. 5–32 (2001) doi: 10.1023/A:1010933404324

2. Byers, S., Wiles, M., Sadie, D., Taft, R.: Mouse estrous cycle identification tool and images. PloS one, vol. 7, pp. e35538 (2012) doi: 10.1371/journal.pone.0035538
3. Clausi, D.: An analysis of co-occurrence texture statistics as a function of grey level quantization. Canadian Journal of Remote Sensing, vol. 28 (2002) doi: 10.5589/m02-004
4. Cora, M., Kooistra, L., Travlos, G.: Vaginal cytology of the laboratory rat and mouse: Review and criteria for the staging of the estrous cycle using stained vaginal smears. Toxicologic pathology, vol. 43 (2015) doi: 10.1177/0192623315570339
5. Doan, M., Case, M., Masic, D., Hennig, H., McQuin, C., Caicedo, J., Singh, S., Goodman, A., Wolkenhauer, O., Summers, H., et al.: Label-free leukemia monitoring by computer vision. Cytometry. Part A : the journal of the International Society for Analytical Cytology, vol. 97, no. 4, pp. 407–414 (2020) doi: 10.1002/cyto.a.23987
6. Gómez-Flores, W., Sossa, H.: Smooth dendrite morphological neurons. Neural Networks, vol. 136, pp. 40–53 (2021) doi: 10.1016/j.neunet.2020.12.021
7. Hemaanand, M., Kumar, V., Karthika, R.: Smart surveillance system using computer vision and internet of things. Journal of Computational and Theoretical Nanoscience, vol. 17, pp. 68–73 (2020) doi: 10.1166/jctn.2020.8631
8. Hernández Hernández, G., Delgado Toral, L., Ochoa Montiel, M. R., Zamora Gómez, E., Sossa, H., Barreto Flores, A., Ramos Collazo, F., Reyes Luna, R.: Estrous cycle classification through automatic feature extraction. Computacion y Sistemas, vol. 23, no. 4, pp. 1249–1259 (2019) doi: 10.13053/cys-23-4-3095
9. Hubscher, C., Brooks, D., Johnson, J.: A quantitative method for assessing stages of rat estrous cycle. Biotechnic & histochemistry : official publication of the Biological Stain Commission, vol. 80, pp. 79–87 (2005) doi: 10.1080/10520290500138422
10. Kaur, S., Benton, W. L., Tongkhuya, S. A., Lopez, C. M., Uphouse, L., Averitt, D. L.: Sex differences and estrous cycle effects of peripheral serotonin-evoked rodent pain behaviors. Neuroscience, vol. 384, pp. 87–100 (2018) doi: 10.1016/j.neuroscience.2018.05.017
11. Ochoa-Montiel, R., Olague, G., Sossa, H.: Expert knowledge for the recognition of leukemic cells. Applied Optics, vol. 59, no. 14, pp. 4448–4460 (2020) doi: 10.1364/AO.385208
12. Pantier, L., Li, J., Christian, C.: Estrous cycle monitoring in mice with rapid data visualization and analysis. Bio-Protocol, vol. 9, no. 17, pp. e3354 (2019) doi: 10.21769/BioProtoc.3354
13. Priddy, B., Carmack, S., Thomas, L., Vendruscolo, J., Koob, G., Vendruscolo, L.: Sex, strain, and estrous cycle influences on alcohol drinking in rats. Pharmacology Biochemistry and Behavior, vol. 152 (2016) doi: 10.1016/j.pbb.2016.08.001
14. Sano, K., Matsuda, S., Tohyama, S., Komura, D., Shimizu, E., Sutoh, C.: Deep learning-based classification of the mouse estrous cycle stages. Scientific Reports, vol. 10, no. 1, pp. 11714 (2020) doi: 10.1038/s41598-020-68611-0
15. Szeliski, R.: Computer Vision. Algorithms and Applications (2011)

ISSN 1870-4069

A Novel Data Augmentation Method based on XAI

Tonantzin Marcayda Guerrero Velázquez, Juan Humberto Sossa Azuela

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Robótica y Mecatrónica,
México

{tmguerrerov, humbertosossa}@gmail.com

Abstract. Machine learning systems allow solving a wide variety of problems present both in industry and in everyday life. The increasing available information has made possible the training of these models and reaches a precision that surpasses the performance of a human being when carrying out the same task. However, the large amount of information necessary to carry out the training task of these models is not always available or difficult to obtain. That is why several methods are used to increase that information from existing training data; these methods are called data augmentation methods. The use of data augmentation techniques allows increment artificially the numbers of samples of a small training dataset, improving the performance of a machine learning model in tasks for which the information available is limited. This paper presents a novel data augmentation technique based on explainable artificial intelligence, where new training samples are created from the explanations generated using the explainability method described in [1]. This method generates explanations based on the classification of useful regions found in an image. Using this method, we can improve the model performance and its accuracy on the test dataset.

Keywords: XAI, CNN, data augmentation,explainability, machine learning.

1 Introduction

Data augmentation is a widely used technique to artificially create new samples information from a small training dataset. This data augmentation is done to obtain a greater range of features and information to feed the models we will use, for example, a CNN (Convolutional Neural Network) that is used for image classification or object recognition tasks. In order to obtain this new augmented dataset, talking about images, several techniques could be applied; some of the most common is based on changing the positioning or the color of the image.

Some of these techniques are: horizontal and vertical shift augmentation (HVA), horizontal and vertical flip augmentation (HVF), random rotation augmentation (RRA), random brightness augmentation (RBA), and random zoom augmentation (RZA). Machine learning models provide solutions to many problems with high precision, but this precision is could due to the quality of information available.

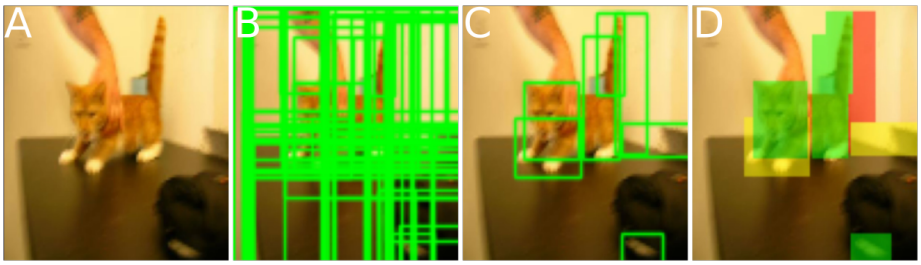


Fig. 1. Generation of the visual explanation of the prediction of an image classifier model. (A) Original image, (B) Set of Candidate regions, (C) Set of useful regions (D) Visual explanation.

However, some cases do not have much information, and the available data is limited, or it is not so easy to obtain; such is the case of the analysis of medical images [2]. Then, it is beneficial for these cases to apply data augmentation techniques. New data augmentation techniques have emerged to solve the problem of having small training dataset. Such is the case of the work described in [3] where they present a new technique based on randomly cutting out regions of four different images and putting them together to form a new image.

This new image will be part of the training dataset, and its label will be proportionally composed of the classes to which the cropped images belong. In [4] a technique called Smart Augmentation is described, which is based on creating a new neural network that learns how to generate new data during the training process of the base model. Another interesting technique is the one described in [5], which uses a set of functions to combine different input images to create a new image. This new image has an unrealistic appearance for a human being; however, according to the results presented by the authors, this technique generates good results for increasing data.

More recent work is that described in [6], where a technique is proposed that is based on randomly selecting a rectangular region of the image and erasing its pixels with random values during the training process. The author reports that with his methodology a reasonable improvement was obtained in the detection of objects as well as in the identification of people. In [7]a data augmentation method is proposed based on producing new information using the explanations generated by applying various existing explainability methods. Its objective is to try to align the predictions obtained from a model with the explanations resulting from the applied explanatory method.

2 Data Augmentation

Data Augmentation methods seek to artificially increase the original training dataset by applying different transformations to this data. In [8] two categories are described that encompass this type of transformation: geometric and photometric. Geometric transformations refer to those that alter the geometry of the image, moving each of its pixels in a new direction. On the other hand, the photometric or color transformations refer to those that alter the value of the RGB channels by changing each pixel value (r, g, b) to a new value (r', g', b') according to the predefined heuristics [8].

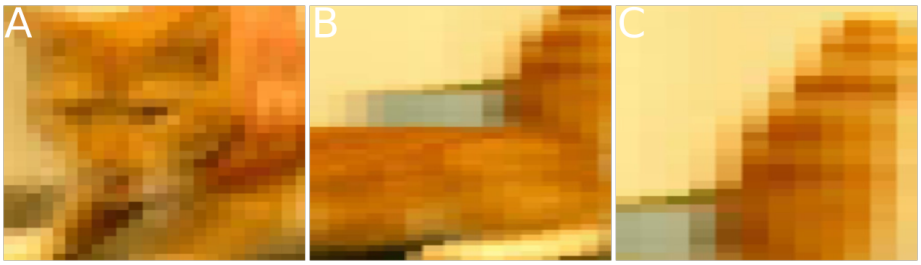


Fig. 2. Example of augmented data generated by applying the explainability method for Figure 1(A).

The data augmentation technique seeks to improve the performance of an image classifier where the amount of information in the training dataset is small. One of the problems with a small dataset is that the model does not generalize adequately for data from the test and validation set, and it is easy that presents a model overfitting problem. With the use of data augmentation, it is possible to reduce this overfitting [9].

Data augmentation is not the only one used to reduce the overfitting of a model. There are other techniques such as Dropout, Batch normalization, Transfer Learning, and Pretraining. However, unlike these techniques, data augmentation deals directly with the root of the problem, that is, the lack of data in the training dataset, as is show in [2].

3 Explainable Artificial Intelligence XAI

Today, machine learning models and deep learning models are widely used to solve complex problems and automate processes, both in industry and everyday life. These models reach surprising levels of precision and good performance, which have often exceeded the results obtained by a human being. The fact that they generate such good results can lead to some mistrust in them since, generally, these models are treated as a black box, of which only the inputs and outputs obtained when processing the information provided are known.

Because these models are used in critical areas such as medicine, medical care, autonomous vehicle management, credit approval, legal and justice issues, etc... , the need arose to create techniques that generate confidence in these models and the results they generate. These techniques are used in order to know the reason for their decisions and how they arrive at such precise results.

They are included in the so-called Explainable Artificial Intelligence (XAI) that arises from satisfying this need and seeks to generate methods that allow understanding the predictions of AI models as well as trying to justify their decisions. There are many explanation techniques that, when are applied to the models, allow us to understand the reason for their results. These methods can be used on different types of data, including images. The methods that provide a visual explanation of the prediction denote areas of the image regions that are of interest to the model when carrying out its prediction so that the explanation can be easily observed.

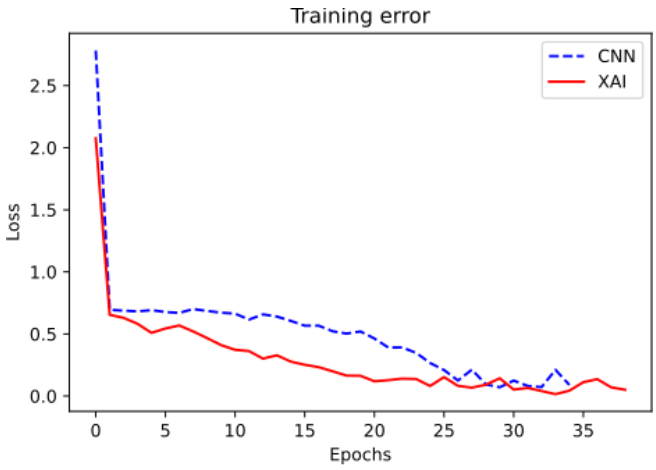


Fig. 3. Training error with and without data augmentation XAI.

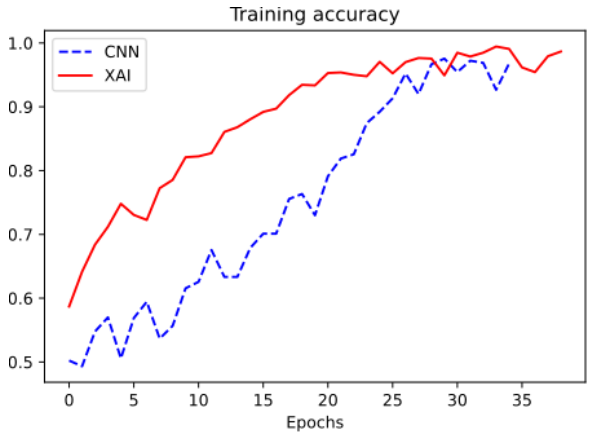


Fig. 4. Training accuracy with and without data augmentation XAI.

In the present work, a new Data Augmentation technique based on XAI is proposed. The augmented data is obtained from the results of the explanations obtained with a explainability method based on the classification of useful regions of an image. The implementation details of the proposed explainability technique are described in [1].

4 Proposed Method

The explainability method [1] in which this paper is based is used for image classifiers and obtains the visual explanation of the prediction of an image classifier model by identifying some regions of the input image (Figure 1(A)) that turn out to be most important for the prediction of the model, called useful regions.

Table 1. Results on the test dataset with and without data augmentation XAI.

Model	Training ACC	Validation ACC	Testing ACC	AUC
CNN	0.9278	0.7700	0.6000	0.786
XAI	0.9789	0.7800	0.6500	0.8371

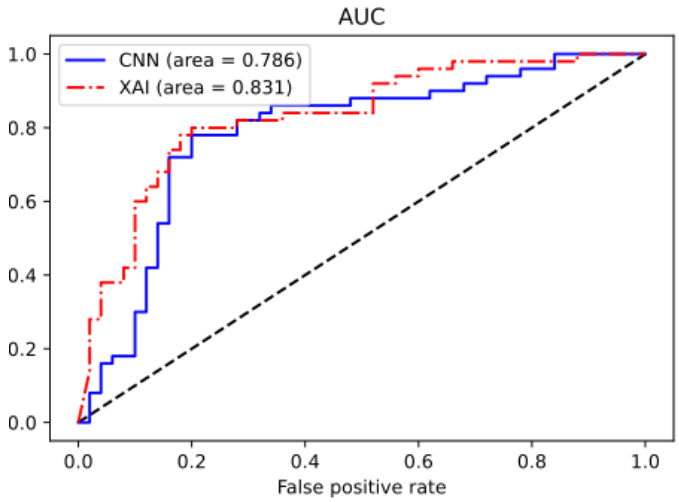


Fig. 5. Curves ROC comparison.

These regions are later statistically categorized, as significant, relevant, or futile and highlighted with green, yellow, and red colors, respectively, to get a visual explanation. The significant regions are marked in green, as can be seen in Figure 1(D). To obtain this explanation, the first thing that is done is a selective search for regions in the input image in order to obtain a set of candidate regions as show in Figure 1(B), so-called because it is not yet known whether they are important to the classifier.

For this reason, each of the regions of this set of candidate regions is evaluated with the classifier and subsequently subjected to statistical analysis to select the really important regions, which are called useful regions which are denoted in Figure 1(C). Finally, this set of useful regions is categorized and colored according to their importance in significant, relevant, or futile, as we can see in Figure 1(D). With this, the level of importance that each of these regions represents for the classifier model is denoted at the time of carrying out its prediction.

The details of this process are described in our work cited in [1]. Figure 1 shows the hole process to obtain a visual explanation for an image, using a Cat vs Dog classifier. The proposed Data Augmentation technique is based on this explainability method [1] and consists of adding to the original training dataset all those regions categorized as significant (black regions in Figure 1(D)) that result from the explanation obtained for each of the data in the training dataset that were correctly classified by the model.

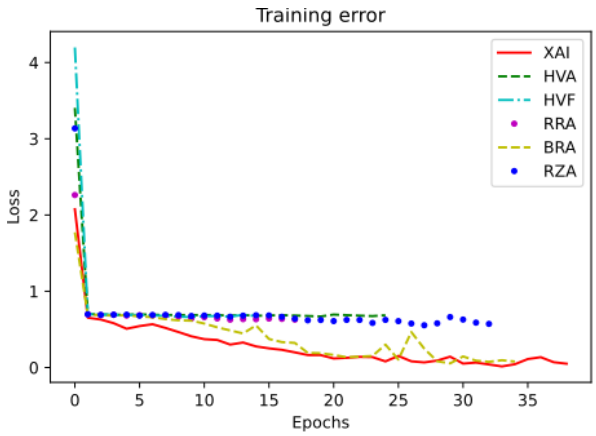


Fig. 6. Training error data augmentation methods comparison.

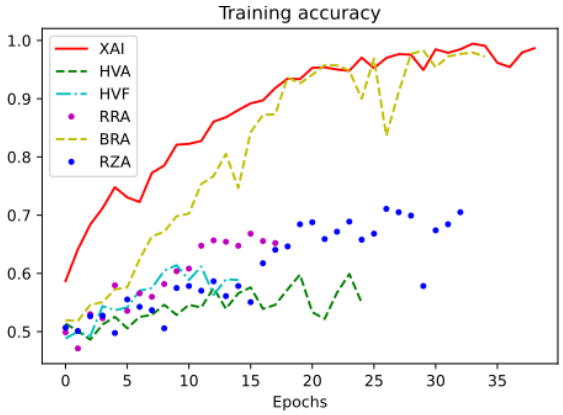


Fig. 7. Training Accuracy data augmentation methods comparison.

Figure 2 shows data that belongs to the augmented dataset generated by applying the explainability method to the original 100 x 100 pixels image. Then the created new samples are the regions identified with the explanation of the training dataset predictions and using the significant regions, it can be ensured that relevant information will be provided to the model, increasing the performance and accuracy of the model.

5 Results and Discussion

In the present work, we use the dataset Dogs vs. Cats that was taken from Kaggle [10], which contains a total of 25,000 images for the training dataset and 12,500 images for the test dataset. However, only a subset of 1000(100x100 pixels) images was taken from the total, in order to simulate a very small dataset.

Table 2. Comparison between the accuracies of the different methods.

Model	Training ACC	Validation ACC	Testing ACC	AUC
XAI	0.9789	0.7800	0.6500	0.831
HVA	0.6467	0.6900	0.6600	0.707
HVF	0.6289	0.6600	0.6200	0.685
RRA	0.6589	0.7000	0.6250	0.714
BRA	0.9644	0.8000	0.6450	0.846
RZA	0.7256	0.7800	0.6700	0.798

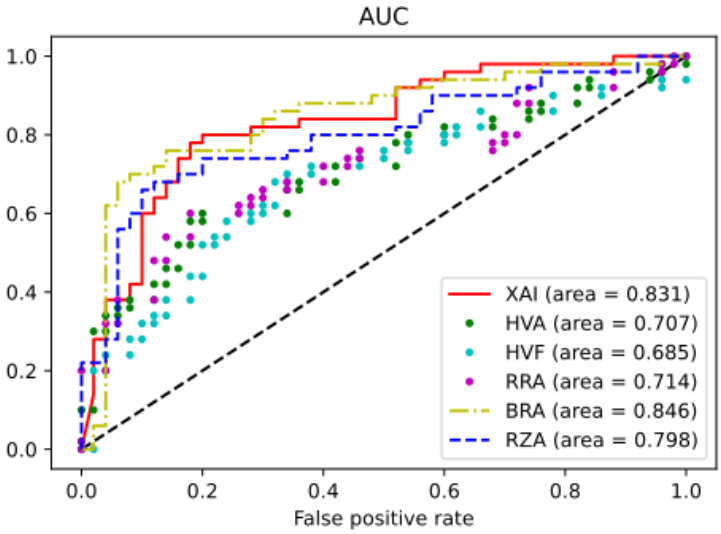


Fig. 8. AUC comparison.

Throughout this work, we use a CNN model with *Adam Optimization* and *categorical_crossentropy* as the loss function. From the 1000 images of the new small dataset, 90% are used for training and 10% for validation. Another different set of 200 images from the remaining 24000 from the original training dataset will be used for testing.

5.1 Data Augmentation XAI Based Method

After much effort, the best result we could get with our CNN model was accuracy of 0.9278 for training and 0.77 for validation, and using data augmentation with XAI as describe before we get an accuracy of 0.98 for training and 0.78 for validation. A comparison between the loss and accuracy evolution in training with and without data augmentation using XAI is shown in Figures 3 and 4. When we compare Figures 3 and 4, and according to the accuracy of the best model obtained, the data augmentation method using XAI proposed here slightly improves the model and its training.

Furthermore, if we use the AUC of the ROC curve as shown in Figure 5, the model trained used data augmentation is a better model, and it will be chosen in a real production environment where even 0.01 of improvement could make a difference. A better improvement is evident when we compare the results over the test dataset where the XAI method for data augmentation shows better accuracy as is shown in Table 1.

5.2 XAI Method Versus Known Methods

It is fair and necessary to compare the method against others commonly used. To do this, we use five different commonly used methods, horizontal and vertical shift augmentation HVA, horizontal and vertical flip augmentation HVF, random rotation augmentation RRA, random brightness augmentation RBA and random zoom augmentation RZA. In Figure 6, a comparison of the training error is shown where we can observe that the XAI method error is always lower than the others, and in Figure 7, we compare the training accuracy curve where the XAI method is also better. When we compare the AUC for the different methods, we found that the model created with the XAI method is one of the best methods, as shown in Figure 8. Table 2 shows a comparison between the accuracies obtained of the different methods of data augmentation on the different datasets.

6 Conclusions and Future Work

It was proven in this work that our proposed method for data augmentation based on XAI is efficient to obtain a better performance for the model than the one that is achieved without increasing data, this is without causing overfitting or reduction in the accuracy; it was also compared with other methods already known, demonstrating be one of the most outstanding. Our method can also be used in conjunction with any of the methods mentioned above to obtain better results, which will be left for future works.

Acknowledgment

Tonantzin Guerrero thanks CONACYT for the scholarship to undertake her doctoral studies.The authors thank the Instituto Politécnico Nacional for the economic support under projects SIP 20200630 and 20210788, and CONACYT under projects 65 (Fronteras de la Ciencia) and 6005 (FORDECYT-PRONACES).

References

1. Guerrero Velázquez, T. M., Sossa Azuela, J. H.: New explainability method based on the classification of useful regions in an image. Computación y Sistemas, vol. 25, no. 4 (2021) doi: 10.13053/cys-25-4-4049
2. Shorten, C., Khoshgoftaar, T. M.: A survey on image data augmentation for deeplearning. Journal of Big Data, vol.6, no. 1, pp. 1–48 (2019) doi: 10.1186/s40537-019-0197-0

3. Takahashi, R., Matsubara, T., Uehara, K.: Data augmentation using random image cropping and patching for deep CNNs. IEEE Transactions on Circuits and Systemsfor Video Technology, vol. 30, no. 9, pp. 2917–2931 (2019) doi: 10.48550/arXiv.1811.09030
4. Lemley, J., Bazrafkan, S., Corcoran, P.: Smart augmentation learning an optimal data augmentation strategy. IEEE Access, vol. 5, pp. 5858–5869 (2017) doi: 10.1109/ACCESS.2017.2696121
5. Summers, C., Dinneen, M. J.: Improved mixed-example data augmentation. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1262–1270 (2019) doi: 10.48550/arXiv.1805.11272
6. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 7, pp. 13001–13008 (2020) doi:10.48550/arXiv.1708.04896
7. Li, R., Zhang, Z., Li, J., Sanner, S., Jang, J., Jeong, Y., Shim, D.: EDDA: Explanation-driven data augmentation to improve model and explanation alignment (2021) doi: 10.48550/ARXIV.2105.14162
8. Taylor, L., Nitschke, G.: Improving deep learning with generic data augmentation. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1542–1547 (2018) doi: 10.48550/ARXIV.1708.06020
9. Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning (2017) doi: 10.48550/ARXIV.1712.04621
10. Microsoft. PetFinder.com: (Dogs vs.cats dataset)
11. Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., Hovy, E.: A survey of data augmentation approaches for NLP (2021) doi: 10.48550/ARXIV.2105.03075
12. Montserrat, D. M., Lin, Q., Allebach, J., Delp, E. J.: Training object detection and recognition CNN models using data augmentation. Electronic Imaging, vol. 10, pp. 27–36 (2017) doi: 10.2352/ISSN.2470-1173.2017.10.IMAWM-163
13. Hernández-García, A., König, P.: Further advantages of data augmentation on convolutional neural networks. In: International Conference on Artificial Neural Networks. Springer, Cham, pp. 95–103 (2018) doi: 10.1007/978-3-030-01418-6_10
14. Kukačka, J., Golkov, V., Cremers, D.: Regularization for deep learning: A taxonomy (2017) doi: 10.48550/ARXIV.1710.10686
15. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, vol. 15, no. 56, pp. 1929–1958 (2014)
16. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448-456 (2015) doi: 10.48550/ARXIV.1502.03167
17. Molnar, C.: Interpretable Machine Learning. A guide for making black box models explainable (2020) https://christophm.github.io/interpretable-ml-book/.
18. Gunning, D.: Explainable Artificial Intelligence (XAI). DARPA (2017)

ISSN 1870-4069

Features Used to Classify UML Diagrams from Images: a Systematic Literature Review

Juan Carlos Suárez Hernández, Ángel J. Sánchez-García, Oscar Alonso-Ramírez

Universidad Veracruzana,
Facultad de Estadística e Informática, Veracruz,
Mexico

{angesanchez, oalonso}@uv.mx, juancasu_900@hotmail.com

Abstract. This study presents the method and the result of a Systematic Literature Review to identify UML diagrams classified in a process of reusing software design artifacts, and the image characteristics used for the classification task. Twenty-one studies were selected and analyzed. In these studies, 3 types of UML diagrams, 2 types of feature approaches, and 8 classifiers were identified. As results, it was obtained that the classified UML diagrams were class diagram, sequence diagram and component diagram. The two approaches found are the predictive characteristics and the rules approach. The classifiers with the highest performance were decision tree J48, Logistic regression, Decision tables, K-Nearest Neighbor (K-NN), Random Forest, Bagging, Support Vector Machine (SVM) and Naïve Bayes.

Keywords: Classification, UML diagram, systematic literature review, software design, features.

1 Introduction

In this new revolution called Industry 4.0, software plays an important role, since it is the way in which all devices interact. Software Design is one of the most important stages in Software Engineering. This stage is characterized by the development of UML diagrams, which are the guide for developers when programming allowing to save time and effort. Just as there is code reuse, a UML diagram of one system or parts of it can be used for the redesign of another system.

Few companies and software designers use UML diagram image repositories to review previous work and be able to reuse those diagrams. Many times, when terms like "UML diagrams" in browsers are searched, search engines show many erroneous results because they search by tags and not by image content. To solve this, there are the UML diagram image classifiers that recognize if the image contains a UML diagram or not. There are some proposals for UML diagram classifiers. For example, in [1] a classifier for class diagrams is presented.

In this classifier, 23 characteristics and 6 classification algorithms were used, among which decision tree J48, logistic regression, decision tables, Random Forest, Support Vector Machine (SVM), and REP tree were found. They report a 96% accuracy rate on

Table 1. Research questions.

Question	Motivation
Q1. - What UML diagrams have been classified using images?	To know the different UML diagrams classified in the literature
Q2. - What image characteristics have been used to classify class diagrams, sequence diagrams, and communication diagrams?	To know the characteristics of the images used to classify UML diagrams
Q3. - What algorithms are used to classify class diagrams, sequence diagrams, and communication diagrams?	To know the algorithms used to classify images that contain UML diagrams

Table 2. Identified keywords and related terms.

Concept	Related terms
UML diagram	
Image	
Classification	Categorization
Characteristic	Attribute, Feature
Algorithm	
Class diagram	
Sequence diagram	
Communication Diagram	

images that were class diagrams and 91% on images that were not class diagrams. Another example is [2], where authors present a tool, which classifies UML diagrams without considering the characteristics of diagrams.

They use the analysis of grayscale histogram, color histogram, number of straight lines and rectangles among others. Such tool has to have a 95 % accuracy analyzing large sets of images reporting less than 1 second per image. This paper is organized as follows: Section II presents the method followed to carry out the Systematic Literature Reviews (SLR). Section III describes the results. Finally, Section IV draws conclusions and exposes the future work.

2 Research Method

To carry out this review, the method proposed in [3] by Kitchenham and Charters was followed, which is a guide used in SE to carry out SLR. This SLR was conducted in three phases: planning, execution of the search and report. The elements of the guide taken in each phase are presented in this section.

2.1 Planning Phase

This phase consists of the following stages: research questions, search strategy, selection of primary studies criteria and quality assessment. Next, each of the phases will be addressed.

Table 3. Search string proposed.

String	Recall	Prec.	%R	%P
Classification AND image AND algorithms AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	0.0015	0.25	0.15%	25%
Image AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	0.0005	0.5	0.05%	50%
Classification AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	0.0005	0.5	0.05%	50%
Classification AND Algorithms AND ("UML diagrams" OR "Class diagrams" OR "Sequence diagrams" OR "Communication diagrams")	**0.0009**	**0.5**	**0.09%**	**50%**

Table 4. Inclusion criteria.

Data Base	Description
IEEE Xplore	www.ieee.org
Springer Link	www.springer.com
ACM Digital Library	dl-acm.org
Science Direct	www.sciencedirect.com

Research Questions

Table 1 shows the research questions with their motivation to identify classified diagrams, image features, and algorithms in UML diagram classification process.

Search Terms

Table 2 shows the keywords and related terms. This search is limited and focused only on certain UML diagrams that are shown in Table 2, since these are the most used diagrams in the software industry.

Search String

With these search terms obtained from the research questions, four different search string were created. A set of papers was selected through a manual search regarding the classification of UML diagrams from different databases.

Once the possible search strings were proposed, the precision and recall were evaluated, the results of which are shown in Table 3. As a result, when observing the precision and recall of each of the options, it was concluded that the best string is the last one in Table 3 (bold).

Search Strategy

The repositories selected for the search can be seen in Table 3. This is due the access and for containing computer and engineering papers. For the selection of primary studies, inclusion and exclusion criteria were established, which can be seen in Table 5 and 6.

Selection Procedure

The selection process is made up of the following stages:

Table 5. Inclusion criteria.

IC	Description
1	Studies published between the years 2011 to 2021
2	Studies written in English
3	The title and / or abstract have at least two search terms.

Table 6. Exclusion criteria.

EC	Description
1	Do not have access to the full text
2	It is a summary, workshop, opinion piece, presentation, book or technical report
3	It is a duplicate research

Table 7. Quality assessment criteria.

ID	Criteria
QAC 1	Does the paper have the objectives of the study established?
QAC 2	Is the research process used defined?
QAC 3	Are the references less than 5 years from the publication of the paper?
QAC 4	Is the methodology used described in detail?
QAC 5	Is the main objective achieved?

- Stage 1. Primary studies are filtered according to IC1.
- Stage 2. The primary studies are removed according to the ECI and EC2.
- Stage 3. The primary studies are filtered according to the ICI2 and IC3 and the primary studies are removed according to the EC3.

Quality Assessment

Table 7 shows the format for the quality assessment, which contains 5 questions that are answered with "yes" or "no" (rated as 1 or 0 respectively). So that each study can have a final score from 0, which means very poor, to 5, which means very good.

2.2 Execution

In this section, the process of executing the SLR will be commented, talking about the selection of studies, the quality evaluation. As it can be seen in Table 8, the papers were reduced as the selection stages of primary studies were applied.

When performing stage 3, it was observed that only 21 primary studies were preserved. Table 9 shows the selected papers by data source. Figure 1 shows that only 5 papers obtained a score of 4, the other 16 papers had ratings between 2 and 3.

3 Results

The answers to the questions proposed in the planning phase will be presented below. In Fig. 2, the primary studies that passed the selection process are shown. By year of publication. It can be seen that there is a growing trend in the analysis of software design diagrams for their classification from the year 2016.

Table 8. Selection process.

Source	Search	Stage 1	Stage 2	Stage 3
IEEE Xplore	10	9	9	9
ACM Digital Library	972	533	439	10
SpringerLink	3229	2019	232	2

Table 9. Primary studies by Source.

Source	Primary studies
IEEE Explorer	[1] [4] [5] [6] [7] [8] [9] [10] [11]
ACM Digital Library	[2] [12] [13] [14] [15] [16] [17] [18] [19] [20]
SpringerLink	[21] [22]

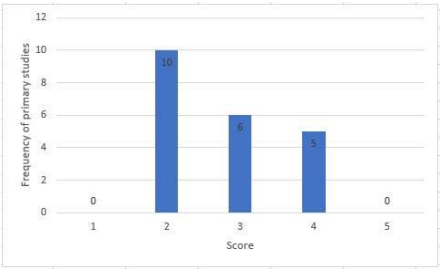


Fig. 1. Scores of quality assessment.

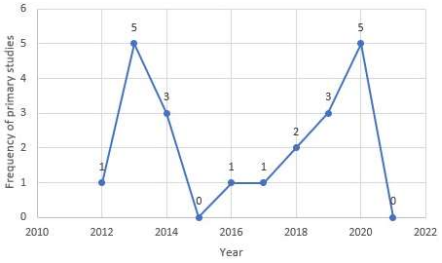


Fig. 2. Primary studies by year of publication.

3.1 Q1.- What UML Diagrams Have Been Classified Using Images

Because of the analysis of the selected papers in the SRL, which resulted in the following more classified diagrams:

- Class diagram
- Component diagram
- Diagram of sequence

As a result, they obtained 95% detection in class diagrams and 91% in non-class diagrams. In [2], authors propose an 8-rule approach to classify class and component diagrams. This approach provided 95% effectiveness classifying UML diagrams, making the best approaches so far.

3.2 Q3. - What Algorithms are Used to Classify Class Diagrams, Sequence Diagrams, and Communication Diagrams?

The algorithms used for classification in this topic are the following: J48 Decision Tree, Logistic Regression, Decision Tables, Random Forest, Support Vector Machine (SVM), REP Tree, OneR, Naïve Bayes, RBF Network, K-Nearest Neighbor (with one and five neighbors), Decision Stump, Random tree and Bagging.

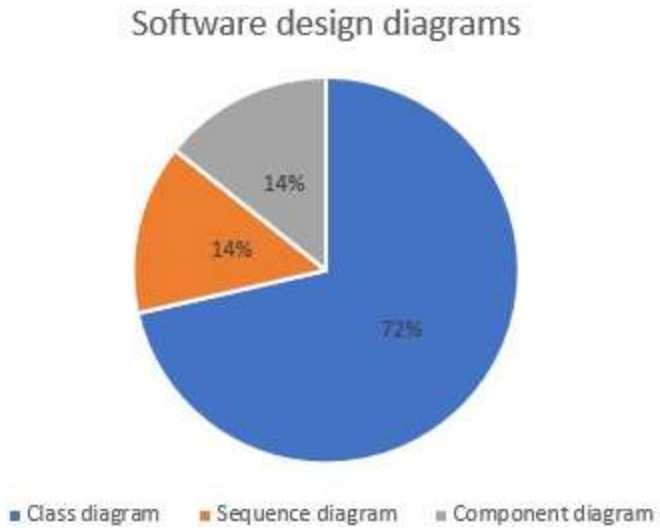


Fig. 3. Reported software design diagrams.

Table 10. Features found in primary studies.

Features in [1]	Features in [2]
Rectangles portion of images, percentage	Number of gray shades
Rectangles size variation, ratio	Number of color shades
Rectangles distribution, percentage	Number of vertical / horizontal segments.
Rectangles connections, percentage	Total number of vertical / horizontal segments at least 30 pixels long
Rectangles dividing lines, percentage	Number of horizontal segments at least 30 pixels long
Rectangles horizontally / vertically aligned, ratio	Number of vertical segments at least 30 pixels long
Average horizontal / vertical line size, ratio	Number of rectangles
Parent rectangles in parent rectangles, percentage	Number of main rectangles (not included in other rectangles)
Rectangles in rectangles, percentage	
Rectangles height-width ratio	
Geometrical shapes portion of image	
Lines connecting geometrical shapes, ratio	
Noise, percentage	
Colour frequency, percentage	

As it can be seen in Fig. 4, the most used algorithms are Decision Tables and the J48 Decision Tree with three appearances, followed by Logistic Regression, Naïve Bayes, OneR and REP Tree with two appearances each.

We must emphasize that, although they are the most used algorithms, they do not necessarily have the best results. Not much detail is given about the selection of the algorithms, more detail is given about the performance they had with the selected features.

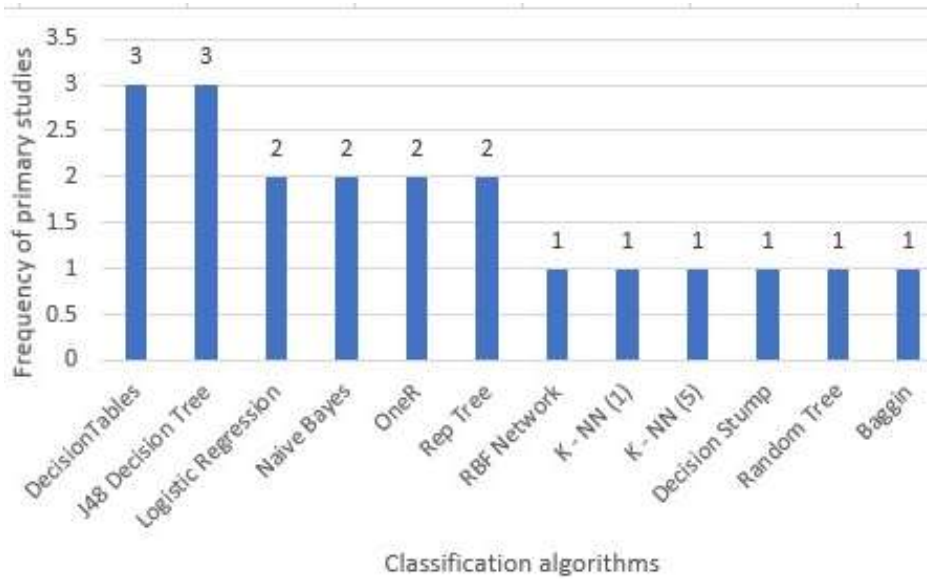


Fig. 4. Classification approaches identified.

4 Conclusions and Future Work

In this work, a Systematic Literature Review on the task of classifying software design diagrams saved in image format was performed and it was based on the method proposed in [3], where the following artifacts were obtained: the search string, the inclusion and exclusion criteria, and the data extraction and quality assessment.

The UML diagrams, when performing the analysis to the data extraction artifacts, the most classified UML diagrams were class diagrams, sequence diagrams and component diagrams.

The best-performing characteristics described by Ho-Quang in [1], as well as Moreno's rule features [2], which will be used together with the UML diagram classification algorithms, to measure performance with the different approaches in our future work.

The classification algorithms such as J48 decision tree, Logistic regression, Decision tables, K-NN (with one neighbor), Random Forest, Bagging, Support Vector Machine, and Naïve Bayes were the best at classifying UML diagram images as reported in [1].

However, we decided to select the following for the next stage of the reception work: J48 decision tree, Logistic regression and SVM. This because they show better results in this review.

Finally, as future work it is intended to extend the functionalities of the work proposed in [2], using a combination of reported algorithms and extending the number of classes in: Class Diagram, Component Diagram, Sequence Diagram and another Diagram.

References

1. Ho-Quang, T., Chaudron, M. R., Samuelsson, I., Hjaltason, J., Karasneh, B., Osman, H.: Automatic classification of UML class diagrams from images. In: Proceedings of Asia-Pacific Software Engineering Conference, vol. 1, pp. 399–406 (2014) doi: 10.1109/APSEC.2014.65
2. Moreno, V., Génova, G., Alejandres, M., Fraga, A.: Automatic classification of web images as UML diagrams. In: Proceedings of ACM International Conference Proceeding Series (2016) doi: 10.1145/2934732.2934739
3. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
4. Osman, M. H., Chaudron, M., Van Der Putten, P.: An analysis of machine learning algorithms for condensing reverse engineered class diagrams. In: Proceedings of IEEE International Conference on Software Maintenance, pp. 140–149 (2013) doi: 10.1109/ICSM.2013.25
5. Godara, D., Singh, R. K.: Improving change proneness prediction in UML based design models using ABC algorithm. In: Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, pp. 1296–1301 (2014) doi:10.1109/ICACCI.2014.6968320
6. Osman, M. H., Ho-Quang, T., Chaudron, M. R.: An automated approach for classifying reverse-engineered and forward-engineered UML class diagrams. In: Proceedings of 44th Euromicro Conference on Software Engineering and Advanced Applications pp. 396–399 (2018) doi: 10.1109/SEAA.2018.00070
7. Jindal, S., Khurana, G.: The statistical analysis of source-code to determine the refactoring opportunities factor (ROF) using a machine learning algorithm. In: Proceedings of IET Conference Publications, pp. 396–403 (2013) doi: 10.104 9/cp.2013.2244
8. Salami, H. O., Ahmed, M.: Class diagram retrieval using genetic algorithm. In: Proceedings of the 12th International Conference on Machine Learning and Applications, vol. 2, pp. 96–101 (2013) doi: 10.1109/ICMLA.2013.112
9. Narawita, C. R., Vidanage, K.: UML generator - An automated system for model driven development. In: Proceedings of 16th International Conference on Advances in ICT for Emerging Regions, pp. 250–256 (2017) doi: 10.1109/ICTE R.2016.7829928
10. Osman, M. H., Chaudron, M. R., Van Der Putten, P., Ho-Quang, T. (2014) Condensing reverse engineered class diagrams through class name based abstraction. In: Proceedings of 4th World Congress on Information and Communication Technologies, WICT 2014, 158–163, (2014) doi: 10.1109/WIC T.2014.7077321
11. Iqbal, S. Z., Huzam, D., Aleliwi, N., Alabdulkareem, A., Alalyani, S., Alfaris, S., Gull, H.: Smart fair guide: requirements and design of a smart fair management system. In: Proceedings of 3rd International Conference on Computer Applications and Information Security, pp. 1–7 (2020) doi: 10.1109/ICCAIS48893.202 0.9096722
12. Lopes, A., Steinmacher, I., Conte, T.: UML acceptance: Analyzing the students' perception of UML diagrams. In: Proceedings of ACM International Conference Proceeding Series, pp. 264–272 (2019) doi: 10.1145/3350768.3352575
13. Balaban, M., Maraee, A.: Finite satisfiability of UML class diagrams with constrained class hierarchy. ACM Transactions on Software Engineering and Methodology, vol. 22, no. 3, pp. 1–42 (2013) doi: 10.1145/2491509.2491518
14. Bian, W., Alam, O., Kienzle, J.: Is automated grading of models effective?: Assessing automated grading of class diagrams. In: Proceedings of 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, pp. 365–376 (2020) doi: 10.1145/3365438.3410944

15. Qi, K., Boehm, B. W.: Detailed use case points (DUCPs): A size metric automatically countable from sequence and class diagrams. In: Proceedings of International Conference on Software Engineering, pp. 17–24 (2018) doi: 10.1145/ 3193954.3193955
16. Saxena, V., Arora, D., Mishra, N.: UML modeling of load optimization for distributed computer systems based on genetic algorithm. ACM SIGSOFT Software Engineering Notes, vol. 38, no. 1, pp. 1–7 (2013) doi: 10.1145/24130 38.2413043
17. Nurwidyantoro, A., Ho-Quang, T., Chaudron, M. R.: Automated classification of class role-stereotypes via machine learning. In: ACM International Conference Proceeding Series, pp. 79–88 (2019) doi: 10.1145/3319008.3319016
18. De Oliveira Barbosa, M., Ramalho, F.: An approach to identify and classify state machine changes from code changes. In: ACM International Conference Proceeding Series, pp. 111–120 (2020) doi: 10.1145/3425269.3425282
19. Mannava, V., Ramesh, T.: Multimodal pattern-oriented software architecture for self-optimization and self-configuration in autonomic computing system using multi objective evolutionary algorithms. In: ACM International Conference Proceeding Series, pp. 1236–1243 (2012) doi: 10.1145/2345396.2345595
20. Relucio, F. S.: Unified modeling and framework design on procurement data standards implementation. In: ACM International Conference Proceeding Series, pp. 126–130 (2020) doi: 10.1145/3379247.3379281
21. Ott, J., Atchison, A., Linstead, E. J.: Exploring the applicability of low-shot learning in mining software repositories. Journal of Big Data, vol. 6, no. 1 (2019) doi: 10.1186/s40537-019-0198-z
22. Best, N., Ott, J., Linstead, E. J.: Exploring the efficacy of transfer learning in mining image-based software artifacts. Journal of Big Data, vol. 7, no. 1 (2020) doi: 10.1186/s40537-020-00335-4

ISSN 1870-4069

Intelligent IoT Platform for Prediction of Failures in the Reception of Equipment Telemetry Requests

Francisco Javier Flores Zermeño, Edgar Gonzalo Cossio Franco

Centro de Investigación y Asistencia Técnica del Estado de Querétaro,
Centro de Tecnología Avanzada, Querétaro,
Mexico

{franc.javier.flores, kofrran}@gmail.com

Abstract. Artificial intelligence and IoT based technologies have generated new opportunities to prevent and identify failures in equipment or machines of all kinds, such as predictive maintenance. This paper presents an implementation of an IoT platform that shows the telemetry of equipment through temperature sensors, magnetic sensors, actuators and voltage detection, which allows the application of statistical and linear regression formulas to analyze the behavior and prediction in the reception of information from the electronic device to the IoT platform.

Keywords: IoT, sensors, telemetry, linear regression, predictive maintenance.

1 Introduction

The Internet of Things (IoT) has revolutionized telemetry as it facilitates the remote monitoring of equipment to observe the behavior of electronic devices in sending alerts to a control center that allows visualization through a web platform [1], in order to supervise that the operation is safe and efficient.

The IoT is increasingly used in the industry and the benefits it offers to companies is increasing justifying the investment required for the implementation of an IoT system, as it guarantees an improvement in the management of resources [2], however, this leads to the maintenance of such devices. There are different types of maintenance; these are classified into three types: corrective, preventive and predictive.

At the beginning, only the term corrective or reactive maintenance was used, which consists of repairing failures as they occur, however over time industries realized the advantage of early detection of a severe failure or the early change of a component with high potential for failure in machines, which gave way to preventive maintenance, which is based on conducting periodic inspections to anticipate the failure and finally periodic monitoring through inspections led to the development of predictive maintenance [3].

Predictive maintenance is based on constantly analyzing the data generated by the equipment to anticipate any possible failure or problem before it occurs, avoiding production downtime and extra costs [4].

The way it is carried out is by defining the normal parameters in which it should operate, and then detecting or predicting alterations or lags in the devices that indicate a possible problem in the system, so the process to define the predictive maintenance of a piece of equipment or device can be subdivided into three stages [5]:

1 The identification and evaluation of maintenance needs.

2 Prioritization of these needs.

3 Formulation of a predictive maintenance program.

Predictive maintenance has been applied to different branches and different techniques have been used for the identification and evaluation of maintenance needs, below, different application models are listed:

1 Fleet management project reporting to the central database where the general details of those equipment are revealed. This research mentions that critical attention should be paid to this area of IoT application so that predictive/preventive maintenance can be performed on equipment and machinery to avoid delays and reduce the adverse effects of carbon emissions to people and the environment [6].

2 Efficient sensor fault management system in IoT smart environments during the operational phase of a building. For effective predictive maintenance planning, machine-learning techniques can be integrated into the workflow, developed to efficiently predict the future condition of individual IoT components, such as data loggers and sensors [7].

3 Prototype that acts as an accessible and interactive option for energy consumption monitoring through BIM (Building Information Modeling), which contributed to predictive maintenance and facilitates the understanding of the condition and performance of building systems [8].

There are techniques that contribute to the prediction of the behavior of a variable such as biased regression or linear regression that seeks to calculate a straight line through the prediction, measuring the error with respect to the sample taken [9]. Through the IoT platform can be collected data and really useful information about the environment and the conditions in which the equipment is, however, also gives us information about the state of the modules that make up the device to achieve predict failures [10].

Therefore, in the proposed project it was determined the development of an IoT platform to receive, store and display telemetry, as for the machine it is necessary to collect data through sensors. This information is very important because it will provide the traceability or historial with which samples are taken from blocks of information and with it to apply statistical models to know the efficiency in the requests. The same sample used previously will be used to apply the linear regression method to predict the behavior in the reception of requests.

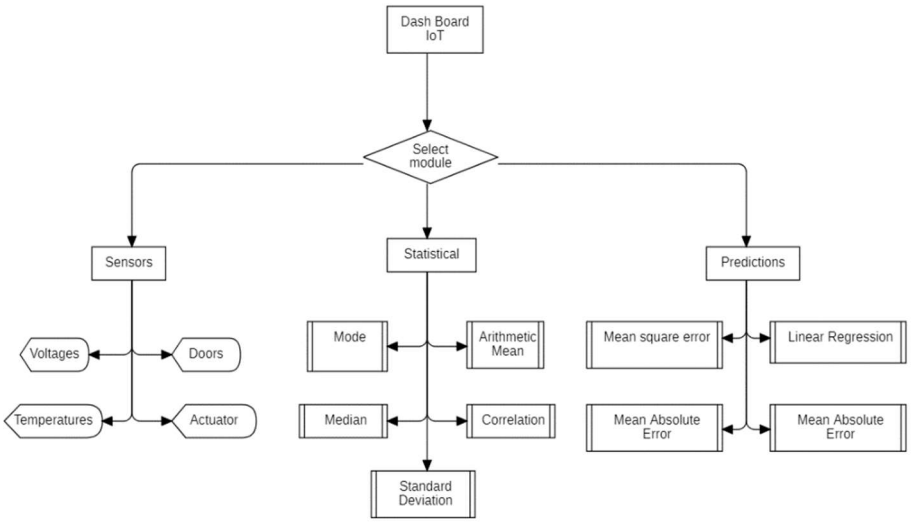


Fig. 1. Structure of the Dash Board of the IoT platform. [Own creation].

2 Methodology

The development of the IoT platform has a visual part that allows the user to access the information in a simple way; a panel of modules, divided into sections, separates the data:

1 Machine telemetry history.

2 Samples collected for the use of statistical methods.

3 Samples for prediction by applying regression method lines and obtaining the prediction data.

In the telemetry part of the equipment, the following data will be obtained: Gate detection, voltage supply, Temperature and Actuators. In the statistics part, the following data will be obtained from the request analysis: Correlation, Arithmetic Mean, Median, Mode and Standard Deviation. In the linear regression part, the prediction will be obtained and the Mean Absolute Error (MAE), the Mean Squared Error (MSE) and the Median Absolute Error (MedAE) will be calculated to check the quality of the prediction. The dashboard diagram includes subdivisions for each module, as shown in image 1.

2.1 Software Description

For the implementation of the IoT platform, we used the Laravel Framework that allows the use of a refined and expressive syntax to create code in a simple object-oriented way [11], this implementation is based on Model-View-Controller (MVC) technology and has a high compatibility for the use of composer packages.

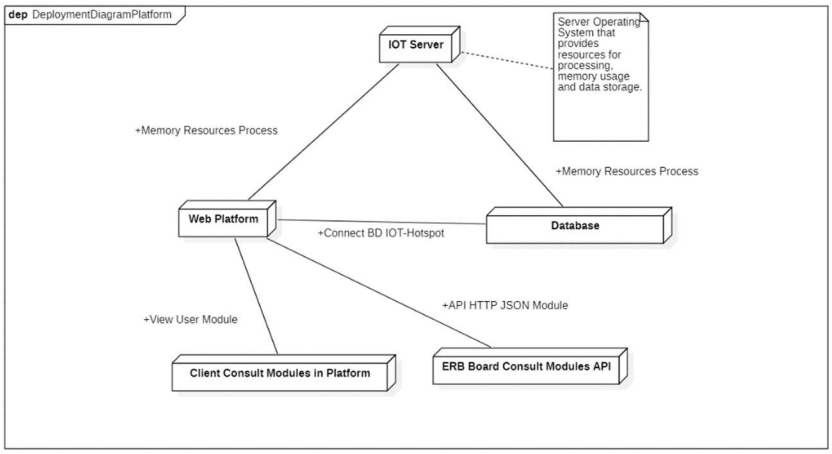


Fig. 2. IoT platform architecture diagram [Own elaboration].

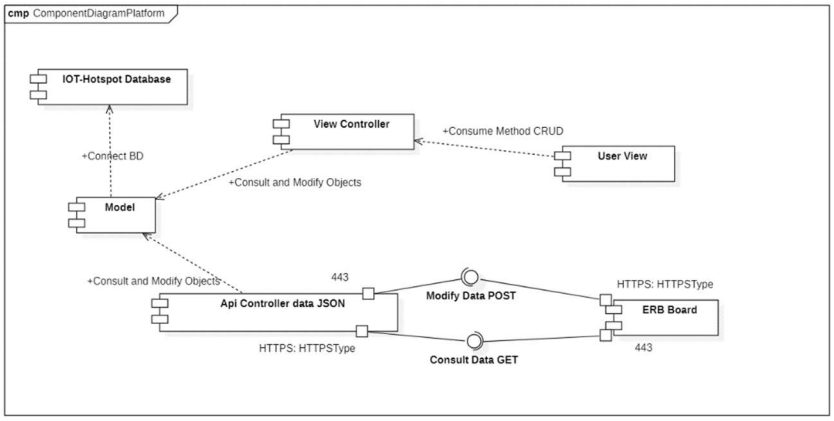


Fig. 3. Diagram of components for the IoT platform [Own elaboration].

In this development it was decided to use the NGINX web server applying a client server type architecture, so for this service the HTTPS communication protocol is used in requests with get() and post() methods for both the visual part of the platform as well as in the rest api.

The telemetry information is sent by the device to the server and then stored in a Maria DB database. This set of configurations is implemented in web services and is known as LEMP SERVER or LAMP SERVER, commonly used for the development of web platforms and in this particular case to develop the IoT platform [12].

2.2 IoT System Architecture

Within the web platform there are two services; the first one is the communication where the Client information is displayed (Client Query Modules in Platform) and the second one is the ERB Card communication (ERB Card Query Modules API).

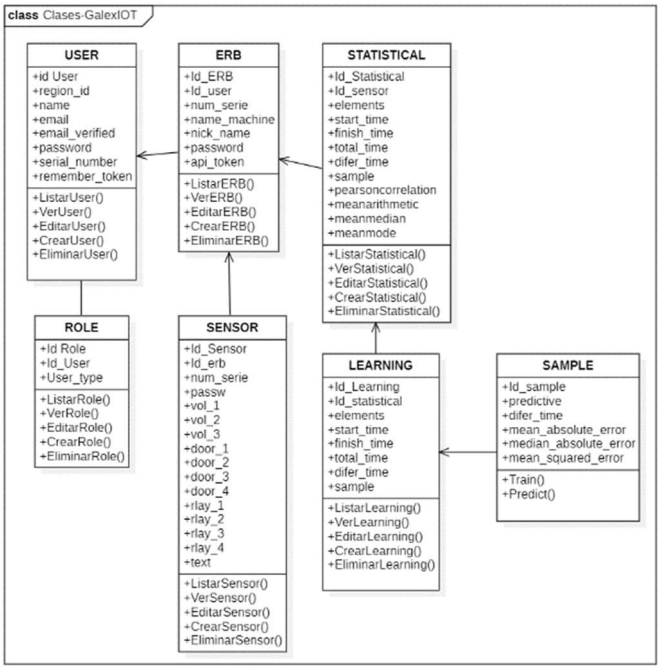


Fig. 4. Class diagram of the IoT platform [Own elaboration].

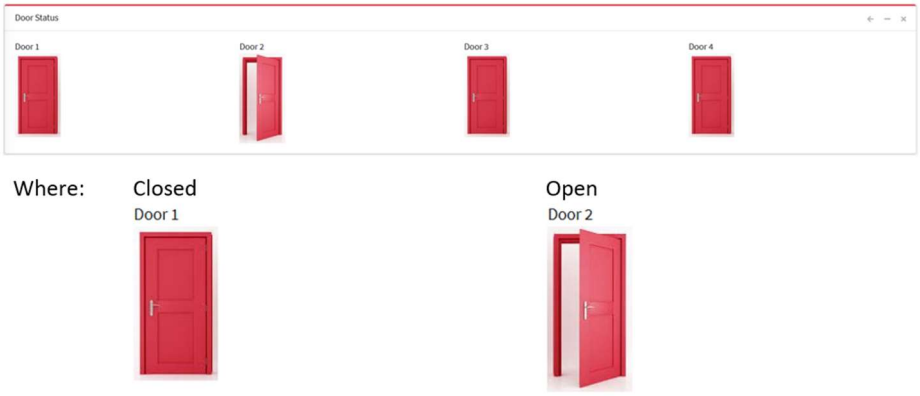


Fig. 5. Door status detector [Own elaboration].

The services pass from the web platform to the database and together they allow saving or querying the information inside the IoT server as shown in image 2. The functionality of the two services mentioned above follows the same pattern. Image 3 shows how the database is related to the model.

The model communicates with the controllers and these in turn communicate with the view, this addressing is triggered by the queries generated either from the user or from the card.

The database has 7 tables, as shown in the class diagram in image 4, which are related

Fig. 6. Voltage detector. [Own elaboration]

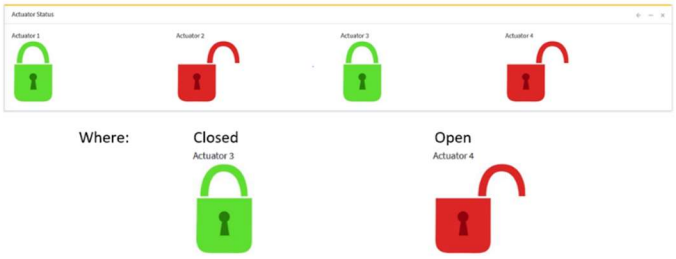


Fig. 7. Status of actuators [Own elaboration]

to each other in order to give integrity to the data of the IoT platform, the relationship of the tables has a unique id and in turn an id foreign.

2.3 Telemetry of the Equipment

Telemetry is a technology that allows the remote measurement of physical variables of a device through data that are transferred to a server [13]. This telemetry was performed by means of wireless communication through get and post requests containing the information of:

- Door status detector: the equipment has two front doors and two rear doors, which must be kept closed, unless the detection of the opening of any of the doors is registered in the device, this is possible through the use of a magnetic sensor that allows the presence or absence of a high pulse to detect that the door is open or a low pulse to detect that the door is closed. On the platform, the detection of the status of the doors is shown as in image 5.
- Voltage detection: the equipment must feed 3 subsystems, which are door power supply, temperature sensors and actuators. Therefore, if no faults are found in any of them, they should be powered all the time. In the platform, voltage detection is shown as in image 6.
- Temperature: the equipment has 4 temperature sensors; these are distributed inside the equipment. The regular temperatures that the equipment supports are from 0 degrees Celsius to 35 degrees Celsius, however, it can reach 65 degrees Celsius and continue working, but from 65.1 degrees Celsius the temperature is considered too high and could cause damage. The graph of the temperature ranges is shown in image 7.

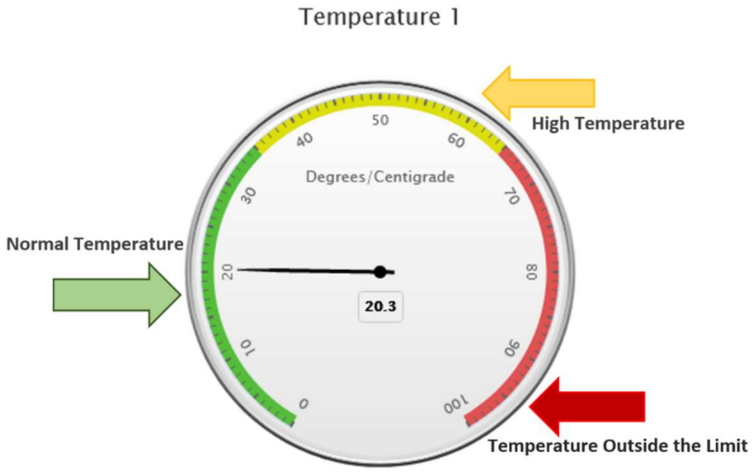


Fig. 8. Temperature graph [Own elaboration].

– Actuators: the units have 4 actuators that are distributed inside the unit. Depending on the equipment, the actuators can be in an open or closed state. On the platform, the detection of the actuator status is shown as in image 8.

2.4 Analysis of Requests

The sending of requests from the equipment to the IoT platform will take 10 minutes; however, there are possible problems that may cause packets not to be sent [14,18], the following is a description of the possible causes:

– Outdated information: if a sensor takes longer than usual, it may delay the sending of requests.

– Slow loading: if the loading takes longer than normal, the server will reject the connection and you will have to try again.

– Interruption in uploads: Due to the slowness, requests may be completely interrupted, taking longer than normal, which will result in the rejection of the request.

– Connection closure: if a connection closure occurs, the equipment will try to recreate the connection and try again.

– Incomplete information: if the information in the get or post request is incomplete, the reception will be rejected and the team will reassemble the information to retry the request.

– Damaged hardware. If there is a physical problem in the hardware that handles network traffic, it is very likely that packet loss will occur.

– Hardware capacity (bottlenecks).

In the analysis of requests, the application of mathematical concepts is contemplated to obtain metrics that help us to understand the data [15], such as:

- Correlation: this metric reflects the measure of association between the number of requests and the time elapsed between them; a correlation close to zero is sought, since it is expected that the values of the samples do not have an increasing or decreasing behavior.
- Standard deviation: is the most common measure of dispersion, indicating how scattered the data are around the mean. A standard deviation value higher than the mean indicates a greater dispersion of the data, implying that the sample values are not similar. In this case the standard deviation will be used to establish a reference value for estimating the overall time variation between requests. For this project, an acceptable standard deviation is considered when the standard deviation is less than 10, taking into account the value should be in 600 with tolerance of ±10.
- Arithmetic mean: it is used as an indicator when the distribution of the sample is normal, which means that it contains a low number of very distant values, which is the ideal.
- Median: it is the midpoint of the sample and indicates the central tendency.
- Mode: is the value that occurs most frequently in the sample and is used along with the mean and median to provide a general characterization of the distribution of the sample data.

If it is considered that the requests are repetitive in a period of time of every 10 minutes which is equivalent to 600 seconds. For this reason, in the ideal case for the project, the arithmetic mean, median and mean should be close to 600, since, if the data are symmetrical, then the mean and median should be similar.

2.5 Linear Regression for Prediction

The prediction or linear regression model is one of the methods used to predict a desired value from a sample [16], in this case it will be used to represent the relationship between the number of the position of the requests and the time that has elapsed between one and another. The model is expressed in a logical-mathematical form and allows to make predictions of the values where the time that has elapsed between one request and another will be taken, starting from the number of the next position of the request.

It is possible to apply the above-mentioned algorithm since there is a linearity in the input data because each request received is expected to have an approximate time of 600 seconds between one request and another, which is equivalent to a request every 10 minutes. The equation used by the linear regression model for training is given by:

$$y = A + B * X\,, \tag{1}$$

The least squares method consists of minimizing the sum of the squares of the errors:

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y_i})^2\,. \tag{2}$$

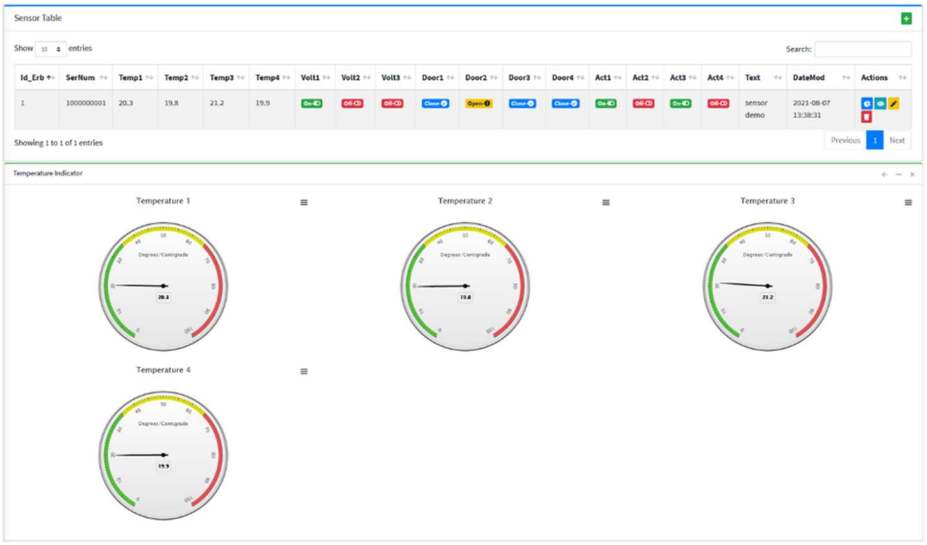


Fig. 9. Status of temperature sensors [Own elaboration].

That is, the sum of the squares of the differences between the actual observed values (y_i) and the estimated values ($\hat{y}_i$). Regression metrics are used to measure the performance of some regression mode and for this project, they are:

- Mean Absolute Error (MAE) will serve to quantify the accuracy of the prediction by comparing the estimated value against the value that is estimated. It is considered a good prediction if it is less than 7 since that represents an error of 1.2% over the desired amount (600), which implies that it is a negligible error.
- Mean Squared Error (MSE) evaluates the quality of the predictions in terms of their variation and the degree of bias, for the purposes of this project it is sought to be less than 41 which represents a negligible error of less than 1.2% over the desired amount (600).
- Median Absolute Error (MedAE) as well as the Mean Absolute Error is considered a good prediction if it is less than 7 since that represents an error of 1.2% over the desired amount (600), which implies that it is a negligible error.

3 Results

The experimentation was carried out under certain conditions that allowed the veracity of the results:

- The conditions of the cloud server were optimal.
- The machines have power supply.
- The machines were free of external agents that could cause failures.
- The internet service was stable for uploading and downloading requests from the machine to the server.

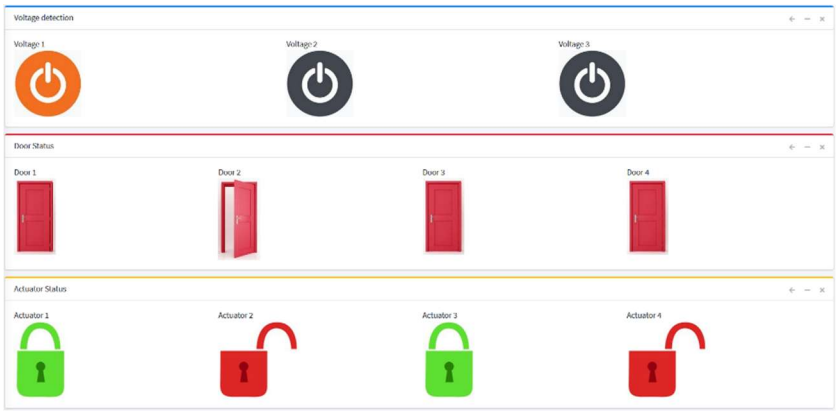


Fig. 10. Status of voltage sensors, gates and actuators [Own elaboration].

Table 1. Petition analysis [Own elaboration].

ID	Correlation	Average Arithmetic	Median	Mode	Standard Deviation
1	0.057291	602	600	605	6.187339
2	0.116996	598	600	602	5.521691
3	-0.083445	595.5	600	595	5.919587
4	-0.096358	600	600	609	6.114546

3.1 Telemetry of the Equipment

The temperature of the equipment is within the desired parameters, which indicates that they are operating at the appropriate temperatures as shown in image 9. As for the voltage detection, the three sensors are on status, as well as the doors and actuators of the equipment are closed as shown in image 10.

3.2 Analysis of the Requests

Four samples were analyzed, each one with 144 requests, these are equivalent to 24 hours or the equivalent of one day of requests in which it is observed that the arithmetic mean, median and mode are close to 600, which implies that the data are symmetrical, it is also observed that the correlation is close to 0 which indicates that it does not have an increasing or decreasing trend, the standard deviation is within acceptable parameters as can be seen in image 11.

3.3 Prediction of Request Failures

For the demonstration of this project and through the use of 4 samples, each one of 144 requests equivalent to one day, these samples were from consecutive days and the prediction of the following day was generated, the start and end time was taken, and the difference and the total time of the samples were obtained, as shown in image 12.

Table 2. Prediction of request failures [Own elaboration].

Id	Elements	Start time	Final time	Total Time	Difference
1	144	2021-07-24 13:38:12	2021-07-25 13:39:35	86483	83
2	144	2021-07-25 13:36:15	2021-07-26 13:38:28	86533	133
3	144	2021-07-26 13:36:12	2021-07-27 13:30:46	86074	-326
4	144	2021-07-27 13:37:09	2021-07-28 13:32:02	86093	-307

Table 3. Details of the prediction of the request [Own elaboration].

Sample	Start time	End time	Pass time	Differ time	Mean absolute error	Median absolute error	Mean squared error
1	2021-07-24 13:38:12	2021-07-24 13:48:12	600	0	5.4514	5	38.0347
2	2021-07-25 13:36:15	2021-07-25 13:46:15	600	0	4.6458	4	30.9375
3	2021-07-26 13:36:12	2021-07-26 13:46:11	599	1	5.1597	5	35.7569
4	2021-07-27 13:37:09	2021-07-27 13:47:08	599	1	5.3125	5	39.0764

To review the generated metrics, access the sample details, in this case a table was generated with the information of sample number, start time, end time, pass time, Difer time, Mean absolute error, Median absolute error, Mean squared error, as shown in image 13. As shown in Figure 13, the Mean absolute error is less than 7, the Median absolute error is less than 7 and the Mean squared error is less than 41, which implies that the errors are less than 1.2% taking as a reference the expected value of 600, so the predictions obtained are within the acceptable parameters, generating an acceptable prediction.

4 Conclusions

Using an IoT platform is essential to centralize the information collected from the equipment, in such telemetry is important to establish what data needs to be collected, such as the time of receipt of the request and the time that exists between them.

Another important point to take into account is to manage the traceability of the telemetry generating a history, if this information is performed a statistical analysis and a predictive method [17], it can generate valuable information, in this case focused on the reception of requests and thus determined the functionality between the equipment and the IoT platform.

The results obtained show that the development of the IoT platform for the equipment was favorable, demonstrating that it is possible to predict failures in the reception of requests, this helps to schedule maintenance visits to the equipment, preventing possible failures that require equipment replacement and thus avoiding an increase in costs.

References

1. Golondrino, G. E., Alarcón, M. A., Muñoz, W. Y.: Sistema IoT para el seguimiento y análisis de la intensidad de luz en plantas de interiores. Research in Computing Science vol. 149, no. 11 (2020)
2. Gómez, A. P., Cahuich, A. C., Gómez, J. J.: Plataforma de gestión IoT mediante técnicas de industria 4.0 para agricultura de precisión. Research in Computing Science, vol. 149, no. 11 (2020)
3. Ynzunza, C., Izar, J., Larios, M., Aguilar, F., Bocarando, J., Acosta, Y.: Tendencias de la gestión de los activos y el mantenimiento predictivo en la industria 4.0: Potencialidades y beneficios. Revista de Aplicaciones de la Ingenieria, vol. 4, no. 11, pp. 30–43 (2017)
4. Tiddens, W., Braaksma, J., Tinga, T.: Exploring predictive maintenance applications in industry. Journal of Quality in Maintenance Engineering, vol. 18, no. 1, pp. 68–85 (2020) doi: 10.1108/JQME-05-2020-0029
5. Pitt, T.: Data requirements for the prioritization of predictive building maintenance. Facilities, vol. 15, no. 3-4 pp. 97–104 (1997) doi: 10.1108/0263277 9710160612
6. Cheung, W. and Huang, W.: Proposing a framework to assess internet usage in university education: An empirical investigation from a student's perspective. British Journal of Educational Technology, vol. 36, no. 2, pp. 237–253 (2005) doi: 10.1111/j.1467-8535.2005.004.55.x
7. Valinejadshoubi, M., Moselhi, O. y Bagchi, A.: Integrating BIM into sensor-based facilities management operations. Journal of Facilities Management (2021) doi: 10.1108/JFM-08-2020-0055
8. Andriamamonjy, A., Klein, R., Saelens, D.: Sensor handling in Building Information Models. Development of a method and application on a case study, pp. 472–478 (2015)
9. Carrasquilla-Batista, A., Chacón-Rodríguez, A., Núñez-Montero, K., Gómez-Espinoza, O., Valverde, J., Guerrero-Barrantes, M.: Regresión lineal simple y múltiple: Aplicación en la predicción de variables naturales relacionadas con el crecimiento microalgal. Revista Tecnología en Marcha, vol. 29, no. 5, pp. 33–45 (2016) doi: 10.18845/tm.v29i8.2983
10. Vijay-Kumar, E., Chaturvedi, S. K.: Prioritization of maintenance tasks on industrial equipment for reliability: A fuzzy approach. International Journal of Quality & Reliability Management, vol. 28, no. 1, pp. 109–126 (2011) doi: 10.1108 /02656711111097571
11. Ghaffari, K., Mehraeen, M. Kazemi, M., Malekzadeh, G.: A comprehensive framework for internet of things development: A grounded theory study of requirements. Journal of Enterprise Information Management vol. 33, no. 1, pp. 23–50 (2019) doi: 10.1108/JEIM-02-2019-0060
12. Molina, Y. A., Ramírez, S. A., Morales, J. E., Reyes, A. N., Sánchez, R. G., García, I. V.: Diseño y desarrollo de un sistema de monitoreo remoto implementando internet de las cosas. Research in Computing Science vol. 149, no. 11 (2020)
13. Wang, G., Li, Q., Sun, J., Meng, X.: Telemetry data processing flow model: A case study. Aircraft Engineering and Aerospace Technology, vol. 87, no. 1, pp. 52–58 (2015) doi: 10.1108/AEAT-11-2012-0221
14. Palacios, R. H., Balseca, M., Gallo, V., Andrade, N., Pisco, J. C., Marcillo, F.: Mecanismo de control de congestión para transferencias de datos en multicast múltiple. Revista Ibérica de Sistemas e Tecnologias de Informação, no. 30, pp. 62–77 (2018)
15. Kumar, A., Jain, S., Yadav, D.: A novel simulation-annealing enabled ranking and scaling statistical simulation constrained optimization algorithm for Internet-of-things (IoTs). Smart and Sustainable Built Environment, vol. 9, no. 4, pp. 675–693 (2020) doi: 10.1108/SASBE-06-2019-0073
16. Astorga-Gómez, J. M.: Aplicación de modelos de regresión lineal para determinar las armónicas de tensión y corriente. Ingeniería Energética, vol. 35, no. 3, pp. 234–241 (2014)

17. Kaparthi, S., Bumblauskas, D.: Designing predictive maintenance systems using decision tree-based machine learning techniques. International Journal of Quality & Reliability Management, vol. 37, no. 4, pp. 659–686 (2020) doi: 10.1108/IJQR M-04-2019-0131
18. Montemayor-Trejo, J. A., Munguía-López, J., Segura-Castruita, M. Á., Yescas-Coronado, P., Orozco Vidal, J. A., Woo Reza, J. L.: La regresión lineal en la evaluación de variables de ingeniería de riego agrícola y del cultivo de maíz forrajero. Acta universitaria, vol. 27, no. 1, pp. 40–44 (2017) doi: 10.15174 /au.2017.1255

ISSN 1870-4069

Hybrid Geodesic System Architecture for Communication and Tracking Tactical Offline

Griselda Cortés[1], José Rodríguez[1], María Vargas[1], Ernesto Enciso[1], Jacob Ávila[1], Adolfo Meléndez[1], Sergio Vigueras[1], Humberto Sossa[2]

[1] Tecnológico de Estudios Superiores de Ecatepec, Mexico

[2] Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico

{gcortes,jose_rodriguez, maria_vargas, ernesto_contreras, fjacobavila, adolfo_melendez, svigueras}@tese.edu.mx, humbertososa@gmail.com

Abstract. This paper describes the development of a hybrid system under a client-server architecture of multi-Tiers and logical multilayers, development with Nodejs-Express-Angular with Object Relational Mapping and Data Transfer Object with MariaDB. Generally, tracking systems with Global Navigation Satellite System technology are slow without a base architecture and adequate development tools and require an internet connection in some of their stages. The results showed its functionality in different situations applied in problems of aeronautic and terrestrial tracking, guaranteeing a projection of personalized geodetic maps and OpenStreetMap in an agile way, efficient, and secure communication in real-time. The proposed architecture allows native development, integration of new modules, and cross-platform implementation in an easy way. Nowadays, hybrid (air and land) tactical are applied in military and medical communications with tracking in real-time, public or private security of objects and people. The hybrid system can be applied in the syndemic problem, which has caused millions of deaths.

Keywords: Geodesic system, tracking, communication offline.

1 Introduction

Currently, the public safety communications and tactical applications have used the hybrid aerial and terrestrial communication systems because they are fast deployment and large coverage capabilities [1]. For two decades there has been the need to track objects, people, and robots to maintain communication between organizations, centralized and remote control. This allows coordinating different activities in a safe way avoiding the intrusion in the transmitted information [2, 3, 4, 5].

The lack of monitoring during the transfer of a patient by aircraft or terrestrial, the need to provide medical assistance in real-time without an internet connection are frequent problems in the medical area. In addition, the objects tracking (germicidal

Table 1. a) EDLP-SMS Interpretation. b) TCLP Example.

T-Hello everybody	User text message, "Hello everybody"
X-1	Predefined message, "All good", addressed to the controller
N-2	Controller message, operation authorization notification
S-1-ROBOTL20	User message, "Request for authorization" To start operation with invoice "User20"

Sender	Use ID = 10
Addressee	User-Controller-ID=01
Position date	01-abr-2020
Position time	13:00:01 (UTC)
Latitude	20.12345 ° North
Longitude	90.12345 ° West
Altitude	7,000 ft
Speed	123 knot
Azimuth:	45 ° with respect to geographic north
Message	Open text
Message sent	Hello everybody

Table 2. Encrypted DLP-SMS format.

Attribute		Label	No. Chars	Format
ID	1	Sender	2	99
	2	Addressee	2	99
	3	UTC	12	AAMMDDhhmmss
Object-Geoposition (G-User)	4	Latitude	8	+ggddddd
	5	Longitude	9	-gggddddd
	6	Elevation	5	99999
	7	Speed	3	999
	8	Azimuth	3	999
Tactical Chat Link Package (TCLP)	9	kind of message	2	X-
	10	Message sent	76	Hello everybody

robots, ambulances, medicine, and so on), the operation, the permanent communication with its operators, and compliance with mandatory health procedures to reduce infections during syndemic problems are challenges that must be solved in the hospitals. Merrill Singer (1990) defines the syndemic as a set of epidemics with implications at the biological and social levels. The syndemic is a consequence of the social type, which is spread in the world by a virus (COVID-19).

The aim of this paper is to propose a model with multilayer client-server architecture used to develop a multiplatform geodetic application to track the tactical operations of a transport fleet, aircraft, objects, and so on; give technical support and make decisions during the execution of its operations by means of secure data link communication (GSM / GPRS / GPS). Development with Nodejs-Express-Angular with Object Relational Mapping (TypeORM) and Data Transfer Object (DTO) with MariaDB. Telecommunications and cryptography have been an integral part of information

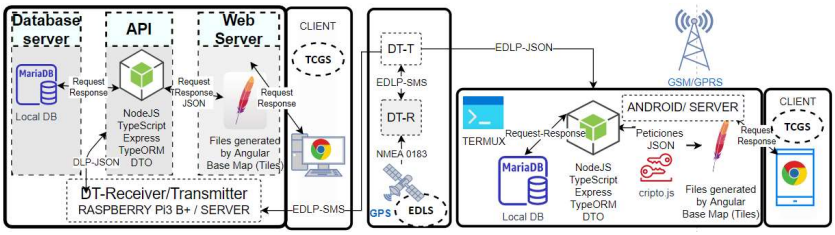


Fig. 1 a) Model of TCGS and EDLS.

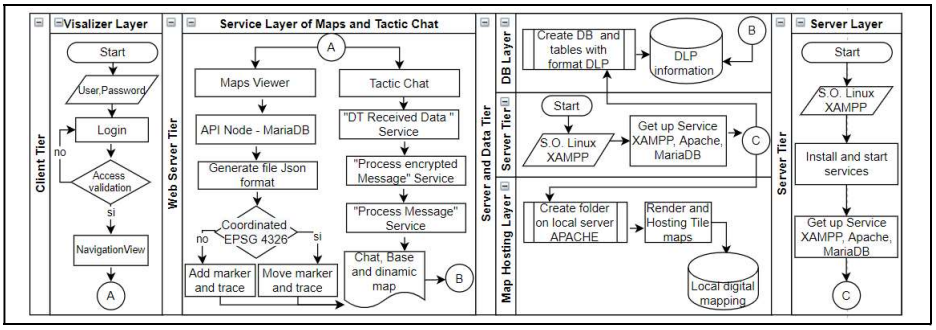


Fig. 2 Tiers and Layers generals of the application.

security, avoiding the intrusion and reception of unwanted or malicious messages [6, 7, 8].

2 Background

In recent years, portable tracking systems, geographic information (GIS), or commercial graphical interfaces [9] have been developed to facilitate the collection, visualization of data on a map [10, 11] and the patterns analysis [12] of objects or people for decision-making. Geographical information [13] is used through digital cartography and recovered from physical sensors such as: Global Navigation Satellite System (GNSS), GPS [2, 14, 15, 16]. GPS digital mapping and geodetics enable geographic data processing and are essential for ground and aerial remote sensing applications. GPS is a space-based radio-navigation system of the United States of America that provides reliable, free, and uninterrupted positioning, navigation, and chronometry services to civil users around the world. It is supported by absolute precision and reliability [17].

China began exploring telemedicine services in the 1980's. Recently, some researchers have developed telemetric systems with Global System for Mobile Communications (GSM) controlled remotely in an encrypted way through SMS [7, 8, 9] and applications Web. This has become a standardized infrastructure for remote monitoring [24, 25].

Related lightweight cryptographic hash functions of the United States National Security Agency and published by the National Institute of Standards and Technology (NIST) provide security on devices with limited resources.

Table 3. TCLP format.

Kind of message	TCLP
Free text[T]	[T-'menssage']. Example: T- Hello everybody
Predefined message [X]	"X-" + key. Example: + [X-1]: all very well + [X-2]: Process started + [X-3]: …
Operation notification [N]	"N" + Event: + [N-0]: operating + [N-1]: Reject + [N-2-'id_Operation']: authorized + [N-3]: Started + [N-4]: Finished + [N-5]: Canceled
Send Operation Request [S]	"S" + Request. Example: + [S-0]: Request of status + [S-1-'Folio of Operation']: Request authorization to start operation

Table 4. Required Disk Space of ECW and Tiles Geomaps.

Geomap	ECW (KB)	Tiles (KB)	Zoom
HP	24,874	131,404,089	7
LP	21,504	134,078,092	7
Relief	431,530	3,285,341,.642	10

Hash algorithms are among the most widespread cryptographic primitives and are currently used in multiple cryptographic schemes and security protocols. They guarantee the integrity and authenticity of the data to achieve a higher level of security.

Boriani and colleagues [26], used the SHA-256 and SHA-512 algorithms with the round pipe technique and obtained a higher yield per cut of 57% and 17%, respectively in comparison to other implementations. S. L. a. K. Shin [16] used SHA-512, which is based on a 32-bit data path. The result is an efficient implementation that can use in IoT security applications. Also, Rote and Selvakumar [27], the authors implemented the SHA-252 hash function in different FPGA families to compare the performance metrics such as area, memory, latency, and clock frequency. This allows a selection of the most suitable FPGA for an application and the implementation of SHA-256 and SHA-512 algorithms.

A decade ago, the N-Tier architecture of physical distribution and n-layers of logical distribution with reusable modules and components emerged in software engineering. Telemetry is a technology that allows the remote measurement of physical quantities and the sending of the information to the operator. This information relates to the data necessary to maintain control of all computer equipment. The operator would be aware of any irregularity or problem that may arise, responding quickly to any mishap, reducing the probability of loss of information or damage to hardware.

Currently, communication can be established from a control center with any portable module to send or receive information and give instructions to operate remotely. In addition, the design and creation of complex systems with maximum scalability,

```
0100210518153125 +1974983 -09901483073800003l4
0100210518153315 +1974983 -0990148207381000314 -T -hello
0100210518153415 +1974982 -0990148207383000314 -T -Operation start
0100210518153819 +1976058 -0989797608657099093 -T -Route start
0100210518154650 +1977832 -0988317409057042131 -T -Heading to Santamaria Ajoloapan
0100210518154750 +1977493 -0988273608811001113 -T -Heading to San Cristóbal
0100210518155158 +1980363 -0988247508687098065 -T -Heading to Santa Maria México
0100210518155209 +1980559 -0988201708637092071 -T -First stop
0100210518155817 +1980644 -0988017408363081013 -T -Second stop
0100210518160251 +1980798 -0987976808392079014 -T -in path
0100210518160626 +1981431 -0987962008199000027 -T -Fourth stop
0100210518161242 +1977898 -0987327508214033267
0100210518163142 +1982360 -0989015808655112257 -T -Fifth stop
0100210518163242 +1981381 -0989308108612109246 -T -Heading to San Pablo
```

Latitude	Longitude	Altitude	Speed	TrackTrue	Time UTC	Satellite
19.533.902	-99.202.227	7253	0.22	102.2	181312.9	5
19.533.908	-99.202.226	7253	0.25	102.2	181313.9	5
19.533.994	-99.202.263	7281	2.49	0.8	181320.0	6
19.534.049	-99.202.315	7308	3.34	270.8	181327.0	7
19.534.144	-99.202.500	7378	0.03	282.3	181338.0	8
19.534.168	-99.202.533	7394	0.58	282.3	181352.1	11
19.535.872	-99.201.157	7421	22.29	94.4	181545.3	12
19.535.276	-99.191.453	7421	8.25	97.2	181734.0	13
19.535.270	-99.191.411	7421	8.45	97.9	181735.1	13
19.535.265	-99.191.368	7420	10.14	97.6	181736.1	13
19.517.191	-99.143.325	7377	24.61	198.3	182749.0	15
19.516.088	-99.143.443	7370	1.98	155	182806.2	15

Fig. 3. a) EDLP-SMS obtained from DT, b) data obtained from Garmin Glo.

reliability, high performance, and integration are required for different applications [17].

On the other hand, the outbreak of the SARS-CoV-2 pandemic has disrupted health, social and economic systems worldwide, giving rise to urgent needs for technical solutions, thus emerging various robotic platforms with an intelligent and autonomous control system [18, 19]. The germicidal SARS-CoV-2 robot used in this project has a real-time tactical communication module and cryptography to ensure bidirectional communication. It has an integrated geodetic web viewer that can monitor a fleet of robots from any device (desktop or mobile) regardless of the S.O. (Windows, Linux, Android). The type of architecture used in this work is another share. The germicidal SARS-CoV-2 robots usually use commercial software and hardware.

3 Proposed Model

The embedded system is shown in Fig. 1. The principal modules are:

Tactical Communication and Geolocation System (TCGS). It is built with N-Layers for the sub-modules: a) *Viewer Interface*. - Shows static layer (base map) and dynamic layer (base map load with markers and routes in real time). b) *Tactical chat interface*. - it communicates with the data tier or server Tier, starts a service to consult user data, manages EDLP-Json of free or predefined text and sends it to the DT. c) E*ncryption* performs the EDLP-Json encryption process with the Hash252 algorithm to avoid intrusions in the client Tier.

Encrypted Data Link System (EDLS). The EDLS has *Data Transceiver (DT)*. have Middleware with technology GNSS and Encrypted, was built on Arduino ATmega328P or Raspberry Pi3 B + 32GB (DT), a data transmission module with GSM / GPRS / GPS technology and antenna (SIM800L and SIM808L, respectively). Its function is to send Encrypted Data Link Package (EDLP-SMS) to georeferenced positions and establish secure communication between DT-Transmitter (DT-T) and DT-Receiver (DT-R). The connection of the DT components is serial with AT commands. The transmission and reception pins (TXD, RXD) are used in a crossway of both circuits and a cellular protocol SIM card, both selected by high sensitivity level (-108dBm, 23dBm). Furthermore, has personalized cryptogram and use 256 bits with standard NMEA-0183.

3.1 Architecture Server-Client with Multi-Layer and Multi-Tier

The embedded system was built with four tiers, and each has *N* logical layers (Fig. 2).

Client Tier. The web application will work on GNU / Linux, and Microsoft Windows; it must be able to authenticate the operator, object, and controller. In addition, it will display the geomaps with relevant information (latitude, longitude, altitude, UTC, and so on) and route of operation in real time. It will Manage operations and data link communication through a tactical chat to provide technical support and make operational decisions.

Web Server Tier. To build a TCGS with the proposed model architecture we require n logical layers to develop API and services, we required: Linux, Nodejs-Express-Angular (Backend, Frontend; respectively), Leaflet plugin to interact with raster and vector maps with EPSG coordinate systems: 4326 and TERMUX: a terminal emulator for Android, it is used to execute Linux commands and install the application on mobiles.

The selection of the tools required a deep analysis on the different map processors: Geoserver, Mapserver, Mapwingis, Luciad, and so on; geographic viewers: Arcmap, Leaflet, Carto.js, D3, Google Maps, Cesium, Node.js, Turf.js, Mapwingis, Luciad, GDAL Python, and so on; and plugins to display dynamic maps in the web browser: Open Layers and Leaflet.

Data Tier. It builds and manages DB with a persistence engine using MariaDB through Object-Relational mapping (TypeORM) and Data Transfer Object (DTO). These tools guarantee database migration in an easy way. Furthermore, we have local map hosting.

Server Tier. Configured with Apache Tomcat to host and manage geomaps in Tiles format at different scales and zoom levels. If they are not in the tiles format, they are preprocessed with Maptiler. It can be hosted on a microprocessor, PC, server or mobile.

Table 2 shows the criteria of the cryptogram “ID / Robot-Position / TCLP” for the security of the data output. The SMS sends 140 characters, and the proposal sends the Encrypted Data Link Package (EDLP) of 45 and 76 chars for predefined or free text messages. Example: EDLP-SMS = "1001201201130001 + 2012345-090123450700013045T-Hello everybody", see Table 1.

ID (1-3) first block with user identifier Sender-Addressee, 100 user’s max; UTC.

G-User (4-8) Latitude: North/South & 7 digits: degrees (0-90), tenths of a degree (0.00001-0.99999)/0.1 arcseconds=3m. Longitude: East/West & 8 digits: degrees (0-180), tenths of a degree / 0.00001 degrees approximately 1m. Altitude: Maximum 99999 ft=30480m, Speed: Maximum 999 knots=1850 km/h. Azimuth: Range 1-360°.

TCLP (9-10) identifies operation geodata, the type of data message sent: a) personalized. - Maximum 76 chars, b) predefined. - they use a key, and they will be hosted in the DB local.

La Table shows the structure of the instructions, operation trace, open message unencrypted or predefined messages. The latter is technical content and just sends a key.

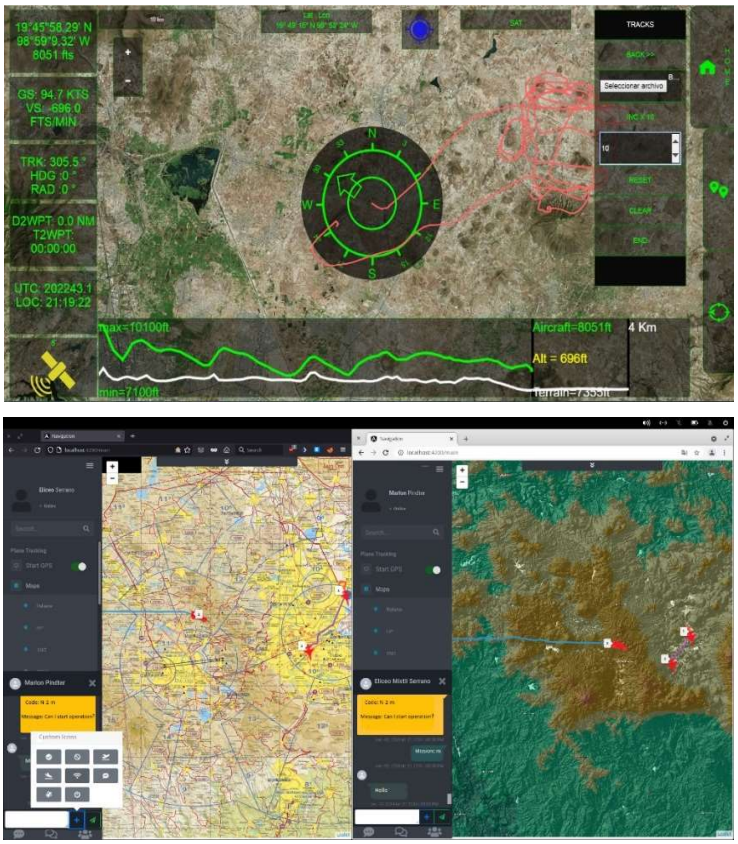


Fig. 4. Visualizer of base geomaps a) with one route b) more of the one route.

Finally, the crypto.js library was used in DLP-Json to continue maintaining the security at the client layer of the data received from the DT-R or DT-T (EDLP-SMS) to the viewer.

4 Results

The proposed EDLS guarantees to send / receive EDLP-SMS, with enough space to send open text messages. The DT allow to generate DLP-SMS of different waypoints and routes (Fig. 3a), the SIM808L module had less data loss, better performance and communication compared to SIM800L and commercial devices such as Garmin Glo (Fig. 3b).

The TCGS visualizes vectorial geomaps (50KT, 100KT, 250KT, 500KT, 1MKT, and so on) with geodetic data offline (Fig. 4b) in an easy way. It was compared with an application that works on Windows 8.1, Tomcat 9, Java, Nodejs, MySQL, GDAL and GeoServer 2.0.8 as a geomaps web Server; It process maps in high compression ratio format, Enhanced Compression Wavelet (ECW), these pre-processed by the Mexican Air Force, Table 4 shows the used hard disk space.

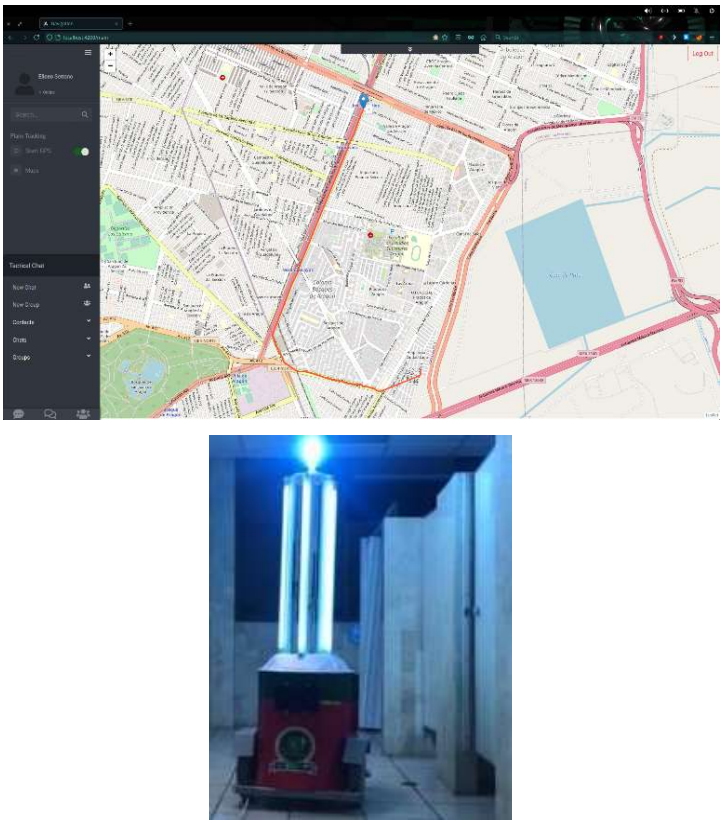


Fig. 5. Tracking of the germicidal Robot.

Unfortunately, the exchange of base layers is very slow, and it crashes when interchanging the dynamic layers or does several processes at a time (Fig. 4a). The proposal is configured on a microcomputer, PC, server or mobile and only requires enough space to host the same base maps in tiles to speed up the work of the cartographic server.

The dynamic maps are successfully graphed, it supports more of one route at the same time, uses custom markers, and can use the tactical chat, all this operates in real- time.

The architecture is validated by installing the application on a HUAWEI Android 8.0 Tablet, 32GB, 256GB; We only require TERMUX and APT package manager for an easy configuration and installation. Also, functionality tests were carried out with waypoints and tracking obtained from an aircraft, achieving its tracking and communication from the control tower in real-time without internet (Fig. 4b).

The same architecture a germicidal robotic platform of the SARS-Cov-2 was integrated, complementing its functionality, supported by OpenStreetMaps. The embedded system allows remote control of the robot, manages, monitors the operations carried out, and provides technical support to the operator in case of failures through tactical chat (Fig. 5).

5 Conclusions and Future Work

This paper achievement the goal and we develop a by using tools opensource, client-server architecture with N-tiler and N-layer, it was developed with innovative tools of high-performance, easy-maintenance, lightweight, robust, and scalable. Furthermore, it solves medical and tactical communication problems (air and land) and tracking of fleets of robots, aircraft, transport, and so on in real time without internet connection, all through GNSS technology. The EDLS achieve to secure, control, encrypt, and guaranteed the predefined or open text EDLP-SMS and permit to send / receive a single packet avoiding the loss or intrusion of this. Also, it maintains security at the client level by using the Hash-512 algorithm and can be installed on any device with GNU / Linux, and Microsoft Windows.

The TCGS was integrate and proof in military aircraft and had good performance. In attention to COVID-19 syndemic to social level, we prevent the spread of the coronavirus, by mean of integration of the TCGS to germicidal robot to track its operation when it to work in hospitals, education institutions, industry, and so on.

The future works it is intended to use the app modules in other projects in any aspect independently, this is possible due to the paradigm of the tools used. On the DT side, it is intended to build a hybrid transceiver (GPS, Radiofrecuency, and Iridium) that always guarantees communication.

Acknowledgments. This work was supported by TESE. H. Sossa thanks the Instituto Politécnico Nacional and CONACyT for the support to undertake this investigation. The authors would like to express their gratitude to the researchers and students for their insightful contributions to attain this research.

References

1. Bucaille, I., Héthuin, S., Munari, A., Hermenier, R., Allsopp, T. R. A. S.: Rapidly deployable network for tactical applications: Aerial base station with opportunistic links for unattended and temporary events ABSOLUTE Example. In: Proceedings of MILCOM´13 IEEE Military Communications Conference, pp. 1116–1120 (2013)
2. Periodico Digital Progresista: El plural, 08 03 2021. [En línea]. Available: https://www.elplural.com/sociedad/ley-normalidad-obligara-mascarilla-airelibre-guarde-distanciamiento-social_262339102. (2021)
3. Santiago-Pérez, M. I., López-Vizcaíno, E., Ruano-Ravina, A., Pérez-Ríos, M.: Sistema de ayuda a la toma de decisiones sanitarias. Propuesta de umbrales de riesgo epidemiológico ante SARS-CoV-2, Archivos de Bronconeumología, vol. 57, pp. 21–27 (2021)
4. Suganya, K., Arulmozhi, V.: Soft computing controller based path planning wheeled mobile robot. In: Proceedings of IEEE International Conference on Advances in Computer Applications (ICACA), pp. 230–234 (2016)
5. Jayasree, K. R., Jayasree, P. R., Vivek, A.: Dynamic target tracking using a four wheeled mobile robot with optimal path planning technique. International

Conference on Circuit, Power and Computing Technologies (ICCPCT), pp. 1–6 (2017)
6. Fahmi, F., Siregar, B., Evelvn, S., Gunawan, D., Andayani, U.: Person locator using GPS module and GSM shield applied for children protection. In: Proceedings of 6th International Conference on Information and Communication Technology (ICoICT), pp. 194–198 (2018)
7. Bada, M., Boubiche, D. E., Lagraa N., Kerrache, C.A., Imran, M., Shoaib, M.: A policy-based solution for the detection of colluding GPS-Spoofing attacks in FANETs. Transportation Research Part A: Policy and Practice, pp. 300–318 (2021)
8. Leguizamon-Páez, M. A.: Sistema para la localización de motos mediante GSM y GPS en Bogotá (2018) [En línea]. Available: https://repository.udistrital.edu.co/bitstream/handle/11349/22420/Frontado%20Escobar%20Camilo%202019.pdf;jsessionid=A129FE4F978B4CC13878D4847FC00348?sequence=4
9. Simik, M. Y. E., Chi, F., Wei C. L.: Design and implementation of a bluetooth-based MCU and GSM for wetness detection. IEEE Access, pp. vol. 7, pp. 21851–21856 (2019)
10. Ali-Arman, M., Tampère, C. M. J.: Lane-level routable digital map reconstruction for motorway networks using low-precision GPS data. Transportation Research Part C: Emerging Technologies, vol. 129, pp. 103234 (2021)
11. Siregar, B., Azmi-Nasution, A. B., Fahmi, F.: Integrated pollution monitoring system for smart city. In: Proceedings of International Conference on ICT For Smart Society (ICISS), pp. 49–52 (2016)
12. Yang, X., Tang, L., Stewart, K., Dong, Z., Zhang, X., Li, Q.: Automatic change detection in lane-level road networks using GPS trajectories. International Journal of Geographical Information Science, vol. 32, no. 3, pp. 601–621 (2018)
13. De Sousa, R. S., Boukerche, A., Loureiro, A. A. F.: Vehicle trajectory similarity: models, methods, and applications. ACM Computing Surveys, vol. 53, no. 5, pp. 1–32 (2020)
14. Nugroho, E. P., Judhie Putra, R. R., Ramadhan, I. M.: SMS authentication code generated by advance encryption standard (AES) 256 bits modification algorithm and one time password (OTP) to activate new applicant account. In: 2016 2nd International Conference on Science in Information Technology (ICSITech), pp. 175–180 (2016)
15. Liu, X., Lu, H., Nayak, A.: A spam transformer model for SMS spam detection. IEEE Access, vol. 9, pp. 80253–80263 (2021)
16. Sang-Hyun, L., Kyung-Wook, S.: An efficient implementation of SHA processor including three hash algorithms (SHA-512, SHA-512/224, SHA-512/256). In: Proceedings of International Conference on Electronics, Information, and Communication (ICEIC), pp. 1–4 (2018)
17. O. U. G. information, GPS.GOV, U.S. Space Force, 18 03 2020. [En línea]. Available: https://www.gps.gov/
18. Serrano-Giné, D., Pérez, Y. M., Àvila-Callau, A., Jurado-Rota, J.: Dataset on georeferenced and tagged photographs for ecosystem services assessment. Ebro Delta, N-E Spain. Data in Brief, vol. 29, pp. 105178 (2020)
19. Sărăcin, A.: Using georadar systems for mapping underground utility networks. Procedia Engineering, vol. 209, pp. 216–223 (2017)

20. Xiao-Tin, H., Bi-Hu, W.: Intra-attraction tourist spatial-temporal behaviour patterns. Tourism Geographies, vol. 14, no. 4, pp. 625–645 (2012)
21. Orellana, D., Bregt, A. K., Ligtenberg, A., Wachowicz, M.: Exploring visitor movement patterns in natural recreational areas. Tourism Management, vol. 33, no. 3, pp. 672–682 (2012)
22. Faka, A., Tserpes, K., Chalkias, C.: Chapter 10 - Environmental sensing: A review of approaches using GPS/GNSS. GPS and GNSS Technology in Geosciences, pp. 199–220 (2021)
23. Hamonangan-Nasution, T., Anggia-Muchtar, M., Siregar, I., Andayani, U., Christian, E., Pascawira-Sinulingga, E.: Electrical appliances control prototype by using GSM module and arduino. In: Proceedings of 4th International Conference on Industrial Engineering and Applications (ICIEA), pp. 355–358 (2017)
24. Shirvani, P., Khajeh-Khalili, F., Neshati, M. H.: Design investigation of a dual-band wearable antenna for tele-monitoring applications. In: Proceedings of AEU International Journal of Electronics and Communications, vol. 138, pp. 153840 (2021)
25. Boriani, G,. De Costa, A., Quesada, A. Prieto-Ricci, R., Favale, S., Boscolo, G., Clementy, N., Amori, V., di S-Stefano, L. M., Burri, H.: Effects of remote monitoring on clinical outcomes and use of healthcare resources in heart failure patients with biventricular defibrillators: Results of the MORE-CARE multicentre randomized controlled trial. Heart Feature, vol. 19, p. 416–425 (2016)
26. Rote, M. D., Vijendran, N., Selvakumar, D.: High performance SHA-2 core using the round pipelined technique. In: 2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1–6 (2015)
27. Ahmad, I., Das, A. S.: Hardware implementation analysis of SHA-256 and SHA-512 algorithms on FPGAs. Computers & Electrical Engineering, vol. 31, no. 6, pp. 345–360 (2005)
28. Tafti, A. P., Rohani, F., Moghaddasifar, M. A.: Towards a scalable G2G framework for customs information system through N-Tier architecture. In: Proceedings of International Conference on E-Learning and E-Technologies in Education (ICEEE), pp. 175–179 (2012)

ISSN 1870-4069

Metodología para un diseño instruccional basado en inteligencias múltiples e IA

Xochitl de Jesus Rojas[1], Carmen Santiago[1], Claudia Zenteno[1], Yeiny Romero[1], Gustavo Rubín[1], Judith Pérez[1], Hermes Moreno Álvarez[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, México

[2] Universidad Autónoma de Chihuahua, Departamento de Ingeniería Aeroespacial Chihuahua, México

{marycarmen.santiago, yeiny.romero, ana.zenteno, gustavo.rubin, judith.perez}@correo.buap.mx, hm1713a@gmail.com,xochitl.de@alumno.buap.mx

Resumen. El proceso educativo tradicional ha dejado de lado aspectos importantes como la forma en cómo aprenden los estudiantes. Por otro lado, la pandemia debido al COVID-19 ha venido a acelerar la incorporación de las TIC al proceso de E-A sin embargo la instrucción mediante materiales en línea aún deja mucho que desear sobre los verdaderos aprendizajes de los estudiantes. En el artículo se propone una metodología para un modelo instruccional que permite ofrecer recursos para el proceso E-A de estudiantes que se adecuan a la forma en cómo aprenden, considerando las inteligencias múltiples y técnicas de IA para medirlas, así como el modelo ASSURE. El resultado es un modelo instruccional inteligente que ha permitido hoy contar con una serie de recursos adecuados para mejorar el rendimiento de los estudiantes de la materia de Metodología de la Programación de Facultad de Ciencias de la Computación.

Palabras Clave: E-Learning, modelo instruccional, inteligencias múltiples, modelo ASSURE, plataformas en línea, inteligencia artificial.

Methodology for an Instructional Design based on Multiple Intelligences and AI

Abstract. The traditional educational process has ignored important aspects such as the way that the students learn. On the other hand, the pandemic due to COVID-19 has accelerated the incorporation of ICTs into the E-A process, however, instruction through online materials is unsatisfactory when we talk about the true learning of students. The article proposes a methodology for an instructional model that offers resources for the E-A process of students that are adapted to the way that they learn, considering multiple intelligences and AI

techniques to measure them, as well as the ASSURE model. The result is an intelligent instructional model that has allowed to have a series of adequate resources to improve the performance of students of the subject of Programming Methodology of the Faculty of Computer Science.

Keywords: E-Learning, instructional model, multiple intelligences, ASSURE model, online platforms, artificial intelligence.

1. Introducción

Al hablar de educación es necesario que tengamos en cuenta que los estudiantes son los que toman el papel principal y la educación no solo debe de tratarse de lo que el alumno es capaz de memorizar o puede hacer, también debe de estar centrada en cómo el estudiante es capaz de recibir el conocimiento, ¿qué puede hacer con dicho conocimiento?, ¿cómo esto le afecta a la sociedad? y ¿cómo le afecta o beneficia asimismo?, por lo tanto se establece lo siguiente: "la educación debe contribuir al desarrollo global de cada persona", sin duda, para lograr alcanzar este objetivo se deben de considerar muchos aspectos para poder tener una educación centrada en el estudiante, algunos de estos aspectos son [1]:

- El alumno necesita un orientador, un tutor y/o un docente.
- Foros donde puede apoyarse para la mejora de su conocimiento.
- El uso de las TIC 's.
- Programas y servicios donde pueda incrementar y poner a prueba su conocimiento.

Si bien es importante considerar estos aspectos, también es importante saber cómo es que el modelo educativo que se ha utilizado hasta ahora es tan plural y esto ha afectado a los estudiantes.

Es claro que, si un estudiante no obtiene los conocimientos suficientes en un sistema de aprendizaje secuencial, en un futuro cuando se vea en la necesidad de utilizar sus conocimientos previos es muy posible que falle, o sea muy difícil para él obtener buenos resultados.

Esto se puede dar puesto que en un sistema educativo en el que 40 alumnos o más comparten un profesor, es muy evidente, que en dicha situación se presente una amplia gama de habilidades y muy poco tiempo para tener o llevar a cabo un enfoque personalizado [2]. Retomando un poco este último aspecto y centrándose en las diferencias de cómo aprenden los estudiantes se debe considerar la teoría de las inteligencias múltiples que son las que cada individuo desarrolla y que van de la mano de cómo aprende.

La teoría de las Inteligencias Múltiples (IM) fue desarrollada por Howard Gardner en el año 1983 y se basa en la idea de que no existe una única inteligencia, sino que ésta tiene múltiples facetas que deben ser cultivadas en las aulas.

Así, el autor identifica ocho tipos de inteligencia diferentes (lingüística, lógico-matemática, visual-espacial, musical, kinestésica, interpersonal, intrapersonal y naturalista). [3] Vea la tabla 1. Ante esto, las IM se define como: "un potencial

Tabla 1. Definición de las 8 Inteligencias Múltiples de Howard Gardner.

Inteligencia	Descripción
Lingüistas	Tienen la capacidad de usar palabras de manera efectiva, ya sea mediante la escritura o de manera oral, por lo tanto, todas aquellas habilidades que tienen que ver con la redacción, y usos del lenguaje son realmente el fuerte de esta inteligencia.
Intrapersonales	Es el conocimiento de sí mismo y la habilidad para adaptar las maneras de actuar a partir de ese conocimiento, se debe de tener una imagen precisa de uno mismo.
Interpersonales	Es la capacidad de percibir y establecer distinciones en los estados de ánimo, las intenciones, las motivaciones, y los sentimientos de otras personas, son sensibles a las expresiones, los gestos y la voz, son buenas sus habilidades para responder de manera afectiva.
Lógico matemática	Es aquella que tiene la capacidad para usar los números, entiende bien los esquemas y relaciones lógicas, las afirmaciones y las proposiciones, se desarrollan de una manera exitosa categorizando, clasificando y generando cálculos.
Corporal-Kinética o Kinestésica	Tienen capacidad para usar su cuerpo para expresarse, tienen facilidad para producir y transformar cosas, también tienen algunas habilidades físicas como la coordinación, equilibrio, destreza y percepción de medidas.
Auditiva o Musical	Es la capacidad de percibir sonidos y formas musicales, tienen sensibilidad al ritmo, tonos, melodías.
Visual	Tienen la habilidad para percibir el mundo visual, espacial, tienen sensibilidad a la temperatura, formas, espacios, relaciones entre los elementos, capacidades visuales y representación gráfica.
Naturalista	Tienen habilidades para la comprensión de la naturaleza, también sus habilidades para la observación, comprensión de hipótesis.

psicobiológico para procesar información que se puede activar en un marco cultural para resolver problemas o crear productos que tienen valor para una cultura".

Si analizamos éstas, observamos la importancia que se otorga al contexto como potenciador de las capacidades de las personas. Gardner no niega el componente genético, pero insiste en que las inteligencias se pueden activar o inhibir en función de las oportunidades que se le ofrecen o se le dejan de ofrecer a un sujeto en cuestión.

Estas oportunidades dependen del ambiente, la educación y la cultura, de aquí la trascendencia de la escuela, la familia y la sociedad en general para que todos los individuos puedan desarrollar al máximo sus capacidades intelectuales.

Considerando la relevancia que la escuela tiene en el proceso educativo y formativo de las personas, resulta evidente que los docentes deben buscar la mejor manera de favorecer el desarrollo integral de sus alumnos. Y no cabe duda de que el hecho de trabajar las IM puede ser una buena estrategia a la hora de potenciar las capacidades de cada uno [4]. Por la parte tecnológica, revisemos que apoyos se tienen para la educación.

Inicialmente hablaremos de IBM EDUCATION el cual nos presenta una tecnología cognitiva la cual ayuda a que profesores y estudiantes mejoren sus resultados, desde el jardín de niños hasta el primer empleo. Con las soluciones y servicios de IBM EDUCATION, se puede personalizar el aprendizaje, aumentar la capacidad de investigación y optimizar las operaciones.

Al conectar estos servicios a sistemas cognitivos (como IBM Watson), crearán un gran avance para la industria, que beneficiará tanto a los profesionales de la educación como a los estudiantes [5].

Los sistemas cognitivos emplean inteligencia artificial (IA) y aprendizaje automático (Machine Learning), por lo que son fundamentalmente diferentes de las computadoras que les precedieron. Mientras los equipos tradicionales deben ser programados por los seres humanos para llevar a cabo tareas específicas, los sistemas cognitivos aprenden de sus interacciones con los datos y los seres humanos, siendo capaces, en cierto sentido, de programarse a sí mismos para llevar a cabo nuevas tareas. En esta era cognitiva, las computadoras se adaptarán a la gente.

Ellas interactúan con nosotros en formas que son más "humanas" [6]. Las modalidades educativas a distancia (EaD) y virtuales, que aprovechan el advenimiento de los dispositivos electrónicos los cuales son cada vez más poderosos y versátiles, junto con el incremento de la capacidad de almacenamiento, organización y recuperación de datos a menor costo, están permitiendo la adopción de los sistemas de educación basados en la Web, sistemas e-Learning (del inglés electronic Learning, aprendizaje electrónico) o Entornos Virtuales de Aprendizaje (EVA).

Los EVA consisten en una colección de herramientas informáticas integradas que facilitan la administración del aprendizaje en línea, generando un mecanismo de entrega, seguimiento del estudiante, evaluación y acceso a los recursos [7]. Los entornos virtuales tienen la gran ventaja de que registran todas las acciones de los usuarios, generando grandes cantidades de datos, que procesándolos permiten realizar el seguimiento del proceso de enseñanza aprendizaje [6].

Éstos sistemas brindan información de cómo son usados, de esta manera, permite el estudio de los datos para hacer correcciones, o permitirles a los usuarios una mejor experiencia en el futuro, así como al profesor le puede ayudar a conocer las debilidades y fortalezas de sus alumnos.

Con el enfoque de la personalización del conocimiento y priorizar la manera en cómo aprende el estudiante, se retoma el análisis de la generación de los contenidos que se le muestran al estudiante.

En el modelo ASSURE se incorporan los eventos de instrucción de Robert Gagné para asegurar el uso efectivo de los medios en la instrucción, este modelo tiene sus raíces teóricas en el constructivismo, partiendo de las características concretas del estudiante, sus estilos de aprendizaje y fomentando la participación activa y comprometida del estudiante. ASSURE presenta seis fases o procedimientos [8]:

- Analizar las características del estudiante: nivel de estudios, edad, características sociales, físicas, etc. Así como conocimientos previos, habilidades y actitudes.
- Establecimiento de objetivos de aprendizaje determinando los resultados que los estudiantes deben alcanzar al realizar el curso, indicando el grado en que serán conseguidos.
- Selección de estrategias, tecnologías, medios y materiales, más apropiado, más adecuados y que servirán de apoyo a los estudiantes para el logro de los objetivos.
- Organizar y desarrollar el escenario de aprendizaje que propicie el aprendizaje, utilizando los medios y materiales seleccionados anteriormente.

- Se requiere la participación de los estudiantes mediante estrategias activas y cooperativas.
- Evaluación y revisión de la implementación y resultados del aprendizaje. Llevará a la reflexión sobre el mismo y a el diseño instruccional para mejoras que redunden en una mayor calidad de la acción formativa.

2. Metodología

El Modelo ASSURE nos permite realizar un planteamiento de un modelo instruccional cuya implementación nos requirió realizar una investigación en distintas plataformas donde actualmente se imparten cursos de diferentes niveles educativos, así como de diferentes áreas, sobre la manera de estructurar los contenidos.

2.1. Análisis de plataformas de e-learning

En el análisis de las plataformas se revisaron cuatro, producto de un sondeo en donde resultaron las más usadas para la capacitación en cursos de TI online por estudiantes de la Facultad de Ciencias de la Computación, las cuales son COURSERA, UDEMY, Tutellus y Platzi. Al realizar este análisis se identificaron los siguientes puntos clave sobre la estructura de los cursos:

- Introducción al curso.
- Material teórico de cada tema- ejemplos prácticos de dicho material.
- Prácticas para realizar, dichas prácticas no se evalúan.
- Evaluación de cada material, evaluación de todo el contenido.

En la mayoría de los casos el material está centrado en 4 puntos principales, los cuales constan de un video introductorio, en el cual se presenta el contenido que se mostrará al alumno, cuales se esperan que sean los conocimientos adquiridos, posterior al material de introducción se muestra material teórico de cada tema, en algunas plataformas se incluyen ejemplos al ver el material teórico, y en algunas otras se muestran ejemplos en un tercer material, el cual retoma algunos puntos a manera de ejemplo, después de ver cierta cantidad de contenido se plantean prácticas, éstas pueden ser a modo de tareas, que si bien no tienen una calificación, se pueden subir a la plataforma para que sean revisadas por otros estudiantes los cuales pueden dejar algún comentario para enriquecer su aprendizaje, estas prácticas también se pueden tomar desde una perspectiva de seguimiento, es decir, el profesor implementa dicha práctica y el estudiante va siguiendo las instrucciones, esto solo para algunas plataformas.

Finalmente hablamos de la evaluación la cual en su mayoría constan de preguntas de opción múltiple y son realmente cortas puesto que no superan los 10 reactivos, estas evaluaciones se realizan al término de un tema, o al término de un bloque establecido de temas.

2.2. Propuesta del modelo para la creación del material

Considerando los 4 puntos clave del análisis de las plataformas y tomando en cuenta el modelo ASSURE, sin olvidar que el estudiante es el elemento principal en el tema de educación, se definen los siguientes elementos que conforman la propuesta del modelo instruccional.

- Material teórico.
- Ejemplos.
- Evaluaciones.

Y es que si bien, el modelo educativo que se ha seguido hasta ahora no es del todo idóneo para todos los estudiantes, con esta propuesta se pretende que el material educativo que se presente se adapte a la manera en la que los estudiantes aprenden.

Para la creación del material teórico inicialmente se considera la evaluación de la inteligencia del estudiante con el objetivo de obtener algunas de las características del tipo de aprendizaje más idóneo para él (vea el Apéndice), como indica el modelo ASSURE la cual nos indica analizar como el estudiante aprende, dicho planteamiento será nuestro punto de partida para el seguimiento a su aprendizaje, el modelo ASSURE resalta que hay que tener en cuenta los objetivos que se pretenden que alcance el estudiante, esto se tiene en cuenta ya que los materiales creados tienen como objetivo cumplir con el contenido del temario de la materia "Metodología de la Programación".

La selección de estrategias que se emplearon en la conformación del material están basadas en algunas de las características más relevantes de cada inteligencia, así como que cada persona no solo maneja una sola inteligencia, sino varias de ellas, finalmente la evaluación de cada una de nuestras estrategias es el punto que nos ayuda a saber si el conocimiento aprendido hasta el momento de la evaluación ha sido exitoso y, en caso de que el estudiante no tenga resultados exitosos se provee aplicarle más preguntas enfocadas a su tipo de inteligencia para así reafirmar que el contenido que le ha sido proporcionado sea el correcto o en su caso analizar la efectividad de éste.

Dicho lo anterior, los materiales se clasificaron en tres grupos atendiendo a las distintas inteligencias y a las características de éstas.

- En el grupo 1, tenemos a las inteligencias lingüista, intrapersonal e interpersonal, éstas se caracterizan por tener una mayor facilidad con la lectura, así como con la extracción o análisis de información, por lo que se les asoció con un material teórico a manera de lectura cuidando que la extensión sea suficiente.
- En el grupo 2, tenemos a las inteligencias musical, visual y naturalista, las cuales tienen mayor habilidad para extraer información a través de materiales visuales como diagramas, videos, películas, etc., y habilidades auditivas, por lo que se les asoció con un material teórico a manera de video, para que tanto musicales, naturalistas como visuales pudieran obtener el conocimiento.
- En el grupo 3, tenemos a las inteligencias kinestésica y lógico matemática, a estos tipos de inteligencias les proporcionamos información a manera de diagramas, puesto que ambas inteligencias tienen grandes habilidades para la experimentación y el análisis de problemas.

Fig. 1 Propuesta de Modelo de Diseño Instruccional.

Tabla 2. Asignación de cada material teórico correspondiente con cada inteligencia.

Inteligencia	Material
Lingüistas	Lectura
Intrapersonales	Lectura
Interpersonales	Lectura
Lógico matemática	Esquema o diagrama
Kinestésica	Esquema o diagrama
Auditiva	Video
Visual	Video
Naturalista	Video

2.3. Modelo de diseño instruccional

El modelo de diseño instruccional que se propone si bien ha sido el resultado de considerar las características y fases propias del modelo ASSURE y del análisis de la estructura de contenido en plataformas de enseñanza, se requiere que inicialmente se realice un test de inteligencias múltiples al usuario y con lo cual obtendremos información sobre cómo el alumno aprende "… capacidad específica de entrada: conocimientos previos, habilidades y actitudes" y "…los medios que serían más adecuados para aprender: texto, imágenes, video, audio, y multimedia", que corresponden a las fase 1 y 3 del modelo ASSURE. Vea Figura 1. Posterior a dicho test, se inicia con la presentación de los contenidos 1. material teórico, 2. Ejemplos y 3. Evaluaciones, en el formato que requieren sus inteligencias más desarrolladas.

2.4. Creación del material educativo

Para la construcción del contenido de un curso/asignatura usando el modelo propuesto, primero se realizó una división de los temas con apoyo de un docente del área quién cuenta con experiencia impartiendo la asignatura, por lo cual todos los temas que engloba el curso de la materia Metodología de la Programación fueron divididos en secciones, con el fin de otorgar gradualmente un material al estudiante y además se comprenda las relaciones entre los distintos conceptos de los temas.

Tabla 3. Material elaborado.

No.	Formato de material
12	Videos formato mp4
11	Diagramas
11	Bancos de preguntas. Con 10 preguntas c/u

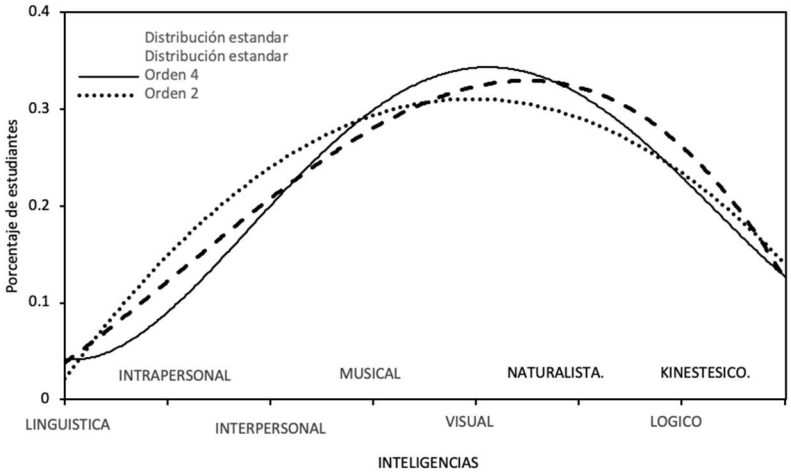


Fig. 2 Resultados del primer test de inteligencias múltiples.

Los contenidos teóricos se crearon en diversos formatos a decir: videos, diagramas/esquemas y materiales de lectura. Este material se generó considerando las inteligencias múltiples y los grupos definidos en 2.2, por lo que en la tabla 2 se muestra la asociación de cada material a las inteligencias.

Es importante mencionar que se diseñaron 3 distintos formatos de materiales teóricos y se cuidó que todos tuvieran el mismo contenido, por lo tanto, un alumno sin importar el formato del material teórico que se le presente será capaz de responder igual que un alumno que aprende con un material de un formato distinto. Dentro del modelo propuesto se considera la evaluación, que corresponde a la fase 6 del modelo ASSURE, evaluación del aprendizaje.

Además, ésta es tomada en cuenta en todas las plataformas analizadas por lo cual se consideró incorporar evaluaciones pequeñas (2 o 3 reactivos) y su realización de manera continua en nuestro modelo, de esta manera y posterior a un material teórico y ejemplos, se realiza una evaluación que permite monitorear dos cosas 1. El aprendizaje del estudiante (si ha aprendido) 2. Evaluar el contenido del tema con vía de mejoras.

Para esta evaluación se construyeron bancos de preguntas, cada banco cuenta con 10 reactivos esperando aplicar reactivos distintos al alumno en caso de fallar en una evaluación. Todos los reactivos cuentan con 4 incisos y el alumno deberá de revisar con detalle para dar una respuesta correcta.

Finalmente, hay que comentar que resultado de esta metodología planteada y modelo de diseño instruccional propuesto le resumimos que, se generaron los materiales listados en la tabla 3 correspondientes a la unidad 1 de la materia propuesta.

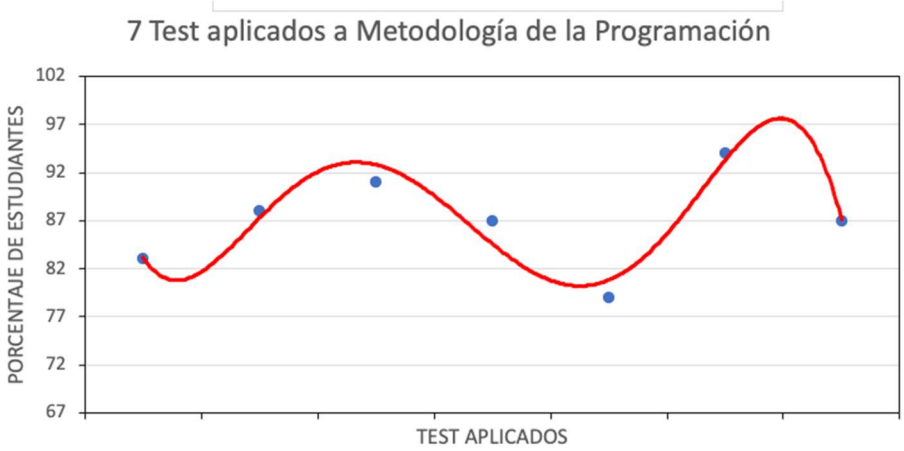


Fig. 3 Resultados de la aplicación de 7 Test de inteligencias múltiples para el máximo "Visual" de la figura 2.

2.5. Determinación de la inteligencia preferente

Inicialmente se aplica una encuesta breve basada en el Test de Inteligencias Múltiples, arbitrariamente se seleccionan 2 preguntas de cada inteligencia y se aplica a cada estudiante de una muestra de 15 estudiantes, a partir de la información que nos brindan, tenemos la gráfica de la figura 2, en la cual hemos utilizado una distribución normal y una regresión polinomial de orden 4, la cual nos genera el mayor índice de correlación con lo cual tenemos un modelo para determinar las probabilidades de la población de estudiantes en el caso de la primera aplicación.

De acuerdo a los resultados de la figura 2 se aplican materiales a los estudiantes con las inteligencias que tienen mayores porcentajes para ellos, aunque la distribución normal para el grupo de estudiantes muestra un máximo para el segundo grupo de inteligencias, se aplica nuevamente en la segunda etapa una vez que el grupo de estudiantes ha concluido la primera actividad y así sucesivamente, con lo cual obtenemos que la probabilidad de estudiantes aumenta alrededor de la inteligencia preferente para el grupo, como se observa en la figura 3, donde nuevamente se utilizaron tanto la distribución normal, como la regresión polinomial de orden 4.

3. Conclusiones y trabajos futuros

En este trabajo enfocamos al estudiante como el actor principal en el proceso enseñanza-aprendizaje, identificando sus inteligencias múltiples y proveyendo de material de aprendizaje adecuado por lo que se espera que tenga una mejor experiencia de aprendizaje y un mayor grado de aprovechamiento en la materia de Metodología de la Programación.

Adicionalmente con ello podremos abrir paso a nuevas oportunidades tanto educativas como profesionales, mejorando sus habilidades ya que cada alumno tiene

una forma de aprender y que probablemente no es la que había utilizado, y por lo tanto los alumnos serán beneficiados al tener una educación dinámica y personalizada.

Se esperan las primeras pruebas de ésta metodología en una plataforma propietaria y con ello procesar las evaluaciones que nos ayude a identificar áreas de oportunidad en el modelo y la metodología propuestas, así como la retroalimentación de la metodología y el proceso de optimización en los recursos didácticos generados.

Referencias

1. Romo, A.: La tutoría. Una estrategia innovadora en el marco de los programas de atención a estudiantes. ANUIES (2022)
2. Spencer, K.: La aplicación de Malawi enseña a los alumnos del Reino Unido 18 meses de matemáticas en seis semanas. BBC, www.bbc.com/news/technology-29063614 (2022)
3. Gorriz B.: Inteligencias Múltiples (2002)
4. Nadal, B.: Las inteligencias múltiples como una estrategia didáctica para atender a la diversidad y aprovechar el potencial de todos los alumnos. Revista nacional e internacional de educación inclusiva, pp. 121–136 (2015)
5. IBM Education. IBM, https://www.ibm.com/mx-es/industries/education, Accedido el 10 de marzo del 2021
6. Urribari L.: Analítica del aprendizaje en un entorno virtual mediante un sistema de computación cognitiva: estudio preliminar. Educación en contexto (2016)
7. Huapaya, C., Lizarralde, F., Arona, G., Vivas, J., Massa. S., Bacino, G., Rico, C., Evans, F.: Uso de ambientes virtuales de aprendizaje en la enseñanza de la ingeniería (2012)
8. Belloch, C.: Modelo ASSURE de Heinich y Col. Universidad de Valencia (2013)

Anexo: Test de inteligencias múltiples

Inteligencia lingüística ¿los libros son muy importantes para mí? ¿Oigo las palabras en mi mente antes de leer, hablar o escribirlas? ¿Me aportan más la radio o unas cintas grabada que la televisión o las películas? ¿Me gustan los juegos de palabras como el Scrabble, el Anagrams o el Password? ¿Me gusta entretenerme o entretener a los demás con trabalenguas, rimas absurdas o juegos de palabras? ¿En ocasiones, algunas personas me piden que les explique el significado de las palabras que utilizo (escritas u orales)? ¿En el colegio asimilaba mejor la lengua y la literatura, las ciencias sociales y la historia que las matemáticas y las ciencias naturales? ¿Aprender a hablar o leer otra lengua (inglés, francés o alemán, por ejemplo) me resulta relativamente sencillo? ¿Mi conversación incluye referencias frecuentes a datos que he leído o escuchado? ¿Recientemente he escrito algo de lo que estoy especialmente orgulloso o que me ha aportado el reconocimiento de los demás?
Inteligencia lógico-matemática ¿Soy capaz de calcular operaciones mentalmente sin esfuerzo?

¿Las matemáticas y/o las ciencias figuraban entre mis asignaturas favoritas en el colegio? ¿Me gustan los juegos o los acertijos que requieren un pensamiento lógico? ¿Me gusta realizar pequeños experimentos del tipo «¿Qué pasará si...?» (por ejemplo, «¿Qué pasará si duplico la cantidad de agua semanal para regar el rosal?»)? ¿Mi mente busca patrones, regularidad o secuencias lógicas en las cosas? ¿Me interesan los avances científicos? ¿Creo que casi todo tiene una explicación racional? ¿En ocasiones pienso en conceptos claros, abstractos, sin palabras ni imágenes? ¿Me gusta detectar defectos lógicos en las cosas que la gente dice y hace en casa y en el trabajo? ¿Me siento más cómodo cuando las cosas están medidas, categorizadas, analizadas o cuantificadas de algún modo?
Inteligencia visual-espacial ¿Cuándo cierro los ojos percibo imágenes visuales claras? ¿Soy sensible al color? ¿Habitualmente utilizo una cámara de fotos o una videocámara para captar lo que veo a mi alrededor? ¿Me gustan los rompecabezas, los laberintos y demás juegos visuales? ¿Por la noche tengo sueños muy intensos? ¿En general, soy capaz de orientarme en un lugar desconocido? ¿Me gusta dibujar o garabatear? ¿En el colegio me costaba menos la geometría que el álgebra? ¿Puedo imaginar sin ningún esfuerzo el aspecto que tendrían las cosas vistas desde arriba? ¿Prefiero el material de lectura con muchas ilustraciones?
Inteligencia musical ¿Tengo una voz agradable? ¿Percibo cuándo una nota musical está desafinada? ¿Siempre estoy escuchando música: radio, discos, casetes o compactos? ¿Toco un instrumento musical? ¿Sin la música, mi vida sería más triste? ¿En ocasiones, cuando voy por la calle, me sorprendo cantando mentalmente la música de un anuncio de televisión o alguna otra melodía? ¿Puedo seguir fácilmente el ritmo de un tema musical con un instrumento de percusión? ¿Conozco las melodías de numerosas canciones o piezas musicales? ¿Con sólo escuchar una selección musical una o dos veces, ya soy capaz de reproducirla con bastante acierto? ¿Acostumbro a producir sonidos rítmicos con golpecitos o a cantar melodías mientras estoy trabajando, estudiando o aprendiendo algo nuevo?
Inteligencia intrapersonal ¿Habitualmente dedico tiempo a meditar, reflexionar o pensar en cuestiones importantes de la vida? ¿He asistido a sesiones de asesoramiento o a seminarios de crecimiento personal para aprender a conocerme más?

¿Soy capaz de afrontar los contratiempos con fuerza moral?

¿Tengo una afición especial o una actividad que guardo para mí?

¿Tengo algunos objetivos vitales importantes en los que pienso de forma habitual?

¿Mantengo una visión realista de mis puntos fuertes y débiles (confirmados mediante feedback de otras fuentes)?

¿Preferiría pasar un fin de semana solo en una cabaña, en el bosque, que en un lugar turístico de lujo lleno de gente?

¿Me considero una persona con mucha fuerza de voluntad o independiente?

¿Escribo un diario personal en el que recojo los pensamientos relacionados con mi vida interior? ¿Soy un trabajador autónomo o he pensado muy seriamente en la posibilidad de poner en marcha mi propio negocio?

Inteligencia interpersonal

¿Soy del tipo de personas a las que los demás piden opinión y consejo en el trabajo o en el vecindario?

¿Prefiero los deportes de equipo, como el bádminton, el voleibol o el softball, a los deportes solitarios, como la natación o el jogging?

¿Cuándo tengo un problema, tiendo a buscar la ayuda de otra persona en lugar de intentar resolverlo por mí mismo?

¿Tengo al menos tres amigos íntimos?

¿Me gustan más los juegos sociales, como el Monopoly o las cartas, que las actividades que se realizan en solitario, como los videojuegos?

¿Disfruto con el reto que supone enseñar a otra persona, o grupos de personas, lo que sé hacer?

¿Me considero un líder (o los demás me dicen que lo soy)?

¿Me siento cómodo entre una multitud?

¿Me gusta participar en actividades sociales relacionadas con mi trabajo, con la parroquia o con la comunidad?

¿Prefiero pasar una tarde en una fiesta animada que solo en casa?

Inteligencia kinestésica

¿Practico al menos un deporte o algún tipo de actividad física de forma regular?

¿Me cuesta permanecer quieto durante mucho tiempo?

¿Me gusta trabajar con las manos en actividades concretas como coser, tejer, tallar, carpintería o construcción de maquetas?

¿En general, las mejores ideas se me ocurren cuando estoy paseando o corriendo, o mientras realizo alguna actividad física?

¿Me gusta pasar mi tiempo de ocio al aire libre?

¿Acostumbro a gesticular mucho o a utilizar otras formas de lenguaje corporal cuando hablo con alguien?

¿Necesito tocar las cosas para saber más sobre ellas?

¿Me gustan las atracciones fuertes y las experiencias físicas emocionantes?

¿Creo que soy una persona con una buena coordinación?

¿No me basta con leer información o ver un vídeo sobre una nueva actividad: necesito practicarla?

ISSN 1870-4069

Investigación sobre técnicas y estrategias de prueba del software: un estudio de mapeo sistemático sobre la última década

Naomi García, Julio C. Díaz, Raúl A. Aguilar

Universidad Autónoma de Yucatán,
Facultad de Matemáticas, Mérida,
México

A17016302@alumnos.uady.mx, {julio.diaz, avera}@correo.uady.mx

Resumen. Pruebas de Software, es un área de conocimiento de la Ingeniería de Software que aborda aspectos vinculados con la mejora de la calidad, en el contexto del proceso. El presente estudio tiene como objetivo ofrecer una visión general de la investigación desarrollada en la última década, con base en las peguntas de investigación formuladas, con la intención de identificar áreas de oportunidad para continuar con estudios primarios o secundarios en temáticas específicas. Se realizó un Estudio de Mapeo Sistemático mediante el cual se seleccionaron 123 estudios primarios, los cuales fueron analizados para identificar la frecuencia y el tipo de investigaciones realizadas en la última década; el estudio plantea el análisis de aspectos como la innovación en cuanto a técnicas de prueba, la frecuencia en las que los estudios mencionan las estrategias y niveles de prueba, así como los mecanismos de evaluación utilizados en el contexto de la investigación empírica. Los hallazgos indican una amplia variedad de factores de calidad analizados, de manera que las técnicas y estrategias varían dependiendo del tipo de software, el área al que se aplica, entre otros factores. El estudio realizado permitió concluir que la investigación en el ámbito de pruebas del software es aún vigente en el contexto de la Ingeniería de Software, por lo que resulta pertinente continuar la investigación en temáticas específicas mediante estudios tanto primarios, como secundarios.

Palabras clave: Calidad del software, pruebas del software, validación, verificación.

Research on Software Testing Techniques and Strategies: A Systematic Mapping Study over the Last Decade

Abstract. Software Testing is an area of knowledge of Software Engineering that addresses aspects related to quality improvement, in the context of the process. The objective of this study is to offer an overview of the research developed in the last decade, based on the research questions formulated, with the intention of identifying areas of opportunity to continue with primary or

secondary studies on specific topics. A Systematic Mapping Study was carried out through which 123 primary studies were selected, which were analyzed to identify the frequency and type of research carried out in the last decade; The study proposes the analysis of aspects such as innovation in terms of testing techniques, the frequency with which studies mention testing strategies and levels, as well as the evaluation mechanisms used in the context of empirical research. The findings indicate a wide variety of quality factors analyzed, so that the techniques and strategies vary depending on the type of software, the area to which it is applied, among other factors. The study carried out allowed us to conclude that research in the field of software testing is still valid in the context of Software Engineering, so it is pertinent to continue research on specific topics through both primary and secondary studies.

Keywords: Software quality, software testing, validation, verification.

1. Introducción

La Ingeniería de Software (IS) dispone desde hace casi un par de décadas, de un cuerpo de conocimientos reconocido por académicos y profesionistas vinculados con la disciplina; la parte medular de la misma integra diez áreas de conocimiento relacionadas con los procesos de desarrollo y de gestión del software, las cuales fueron documentadas en 2004 [1] y actualizadas en 2014 [2]. Una de las áreas de desarrollo directamente vinculadas con el proceso de mejora a la calidad del software, es la de Pruebas de Software; en este sentido, encontrar un proceso de desarrollo software que satisfaga las necesidades reales del cliente y con ello generar software de mejor calidad, ha sido uno de los objetivos de la IS desde el momento de su concepción como disciplina, para ello, los cuestionamientos: (1) ¿Cómo saber si el producto construido funciona correctamente? y (2) ¿Cómo saber si el producto corresponde a lo que el cliente necesitaba? Han tratado de ser atendidos con la investigación vinculada a las técnicas y/o estrategias para verificación y validación del software [3].

De acuerdo con [4] el proceso pruebas tiene como objetivo directo el revelar defectos en el software, y de manera indirecta medir el grado de calidad que posee, con respecto a un conjunto de atributos seleccionados. Una primera postura sobre las pruebas del software establece que estas representan un proceso de verificación dinámica de los comportamientos esperados del software, en función de un conjunto de casos seleccionados [2], y con ello, la no adecuación de dichos comportamientos generaría fallos en el sistema.

Una segunda postura un tanto más flexible considera que hay muchas formas de evaluar, o en su caso, probar un sistema sin requerir ejecutarlo [5]; por ejemplo, las revisiones son un tipo de técnica de prueba que se puede usar para verificar la calidad de un artefacto de software, como puede ser un documento de especificación de requisitos o un listado de código; dicha técnica permite identificar problemas o faltas en dichos artefactos. Ambas posturas, tanto la del análisis dinámico, como la del estático, representan maneras de dar respuesta al primer cuestionamiento antes citado.

Por otro lado, el segundo cuestionamiento que involucra al cliente, es de carácter más subjetivo, y se vincula con el proceso de validación del software, dicho proceso es

definido como aquel que se encarga de la evaluación del software durante o al final del ciclo de desarrollo, para determinar si satisface los requisitos especificados y acordados con el cliente [6].

Durante mucho tiempo se afirmaba que realizar pruebas al software era una actividad que debía ocurrir al final del proceso de desarrollo, no obstante, esta perspectiva ha cambiado en las últimas décadas, en virtud de que se ha demostrado que el realizar pruebas en etapas tempranas del desarrollo evita encontrar defectos en el software al momento de su entrega, permitiendo así, desarrollar un software de mejor calidad.

En relación con el enfoque abordado por la estrategia de prueba, existen dos maneras: (1) Conociendo el funcionamiento interno del software, se pueden diseñar y ejecutar pruebas para asegurar que las operaciones internas se realicen de acuerdo con las especificaciones y que todos los componentes internos se hayan ejercitado adecuadamente; y (2) Conociendo la función especificada para la que se ha diseñado el software, se pueden diseñar y ejecutar pruebas que demuestren que cada función es completamente operativa, al mismo tiempo que se identifican fallos en cada función.

En el primer enfoque las pruebas requieren una vista interna del software y son conocidas como pruebas de caja blanca; mientras que el segundo, las pruebas tienen una visión externa y son conocidas como pruebas de caja negra [7].

Así mismo, en cuanto al nivel de la prueba, es posible distinguir cuatro niveles en función del objeto de prueba: (1) pruebas unitarias, (2) pruebas de integración, (3) pruebas del sistema y algún tipo de (4) pruebas de aceptación [8]. Es importante mencionar que tanto las pruebas de caja negra, como las de caja blanca, pueden clasificarse con base en lo que evalúan, dividiéndose así en pruebas funcionales y no funcionales.

Por un lado, tenemos las pruebas funcionales que han sido mencionado anteriormente (unitarias, integración, sistema y aceptación) y, por otro, tenemos las pruebas no funcionales que evalúan cierto tipo de comportamiento en el software como la compatibilidad, seguridad, estrés, usabilidad, rendimiento, entre otros.

Como se ha mencionado anteriormente, el proceso de realización de pruebas de software se realiza con el objetivo de asegurar la calidad del producto final, y es por ello que esta actividad considera un conjunto de factores de calidad [9] que deben evaluarse como la usabilidad, mantenibilidad, seguridad, portabilidad, eficiencia, capacidad de prueba, flexibilidad, fiabilidad, robusticidad, por mencionar algunas. La evaluación de dichos factores pretende verificar que el software a entregar no sólo cumpla con la parte funcional acordada, sino también, con aquellos requisitos no funcionales acordados con los clientes.

Con la intención de explorar áreas de oportunidad para continuar la investigación en el área de Pruebas del Software, los autores se plantearon el desarrollo de un estudio secundario —un mapeo sistemático de literatura— para identificar y clasificar las principales características de la investigación realizada en el área de pruebas de software en la última década; se acordó analizar estudios primarios publicados entre 2010 y 2021, con la intención de actualizar en cierta medida el estado del arte estructurado en el Swebok [1, 2], pero sobre todo, evaluar la pertinencia de continuar la investigación con un estudio primario, o incluso secundario sobre algún tópico particular —una revisión sistemática de literatura.

En la siguiente sección se describe la metodología utilizada para la realización del estudio reportado; la sección tres presenta las tareas desarrolladas para el planeamiento

del estudio; en la sección cuatro se citan las tareas realizadas para la obtención de los 123 estudios primarios finalmente seleccionados; la quinta sección da respuesta a las preguntas de investigación que conducen el estudio; finalmente, en la última sección se hace un análisis de la información recopilada y se reflexiona en torno al propósito planteado para el estudio.

2. Metodología

Con el propósito de recopilar resultados empíricos publicados a lo largo de los últimos años en el área de pruebas del software, y con ello caracterizar la investigación realizada, se optó por realizar un Mapeo Sistemático de Literatura, conocido también como Estudio de Mapeo (EM); dicho tipo de estudio secundario tiene como objetivo, explorar y proporcionar una visión global sobre un área de investigación, lo que permite identificar nichos de oportunidad para estudios primarios, o en su caso, estudios secundarios más específicos, como pudiera ser una revisión sistemática de literatura; de acuerdo con [10] en los EM es común clasificar los hallazgos obtenidos según algún esquema de clasificación predefinido.

Para el desarrollo del presente estudio se utilizó como referencia la guía propuesta en [11] en la cual se establecen las siguientes tareas:

1 Formulación de las preguntas de investigación: El objetivo principal de los EM es proporcionar una descripción general de un área de investigación e identificar la cantidad, tipo de investigación y resultados disponibles dentro de ella.

2 Búsqueda de estudios primarios: Los estudios primarios se identifican mediante el uso de cadenas de búsqueda en Bases de Datos científicas o navegando manualmente a través de actas de conferencias o publicaciones de revistas relevantes.

3 Selección de artículos relevantes: Se aplican un conjunto de criterios de inclusión y exclusión para determinar la elegibilidad de los artículos primarios que serán analizados.

4 Definición de un esquema de clasificación: Los investigadores revisan los resúmenes y buscan palabras clave y conceptos que permitan identificar un esquema de clasificación.

5 Extracción de datos y elaboración del reporte: Cuando se tiene el esquema de clasificación, los artículos relevantes se clasifican en el esquema, es decir, se lleva a cabo la extracción de datos reales y se procede con análisis de los resultados, presentando las frecuencias de las publicaciones para cada categoría, lo anterior hace posible ver qué categorías se han enfatizado en investigaciones anteriores y, por lo tanto, identificar brechas y posibilidades para investigaciones futuras.

Tabla 1. Categorías identificadas para el análisis de los estudios primarios seleccionados.

Aspecto	Categoría
PI02. Innovación.	Nuevo, Mejora, Existente.
PI03. Estrategia de prueba.	Caja Negra, Caja Blanca, Mixto.
PI03. Nivel de la prueba.	Unitaria, Integración, Sistema, Aceptación.
PI04. Mecanismo de validación.	Caso de Estudio, Experimento, Análisis Comparativo.
PI05. Contexto.	Industria, Academia.

3. Planeación del estudio

La revisión del estado del arte plasmada en el SWEBOK y el análisis de algunos materiales considerados como referencias obligadas para el área de pruebas del software [4, 5, 9], permitieron acumular información suficiente para la formulación de las preguntas de investigación.

3.1. Preguntas de investigación

Las preguntas de investigación que guiaron la realización del estudio son:

- PI01 ¿Cuál es la distribución en la última década de los estudios primarios publicados sobre pruebas del software?
- PI02 ¿Cuáles son los niveles de prueba del software abordados por los estudios primarios en la última década?
- PI03 ¿Qué tipo de pruebas de software han sido las más reportadas en la última década?
- PI04 ¿Cuáles son los factores de calidad que se han evaluado en el software durante la última década?
- PI05 ¿Cuáles son las características de las estrategias utilizadas en las pruebas de software que han sido reportadas en la última década?
- PI06 ¿Cuáles han sido los métodos de validación utilizados en la última década por los estudios primarios vinculados con el área de pruebas del software?
- PI07 ¿En qué contexto han sido desarrollados los estudios primarios vinculados con el área de pruebas del software durante la última década?

La revisión del estado del arte, el análisis de algunos materiales considerados como literatura gris para el área de pruebas del software, y la formulación de las preguntas de investigación, sirvieron de base para la realización de un análisis PICOC [12] del área bajo estudio, dicho análisis permitió identificar un conjunto aspectos clave para asistir en el análisis posterior de los estudios seleccionados.

- Población: Software.

Tabla 2. Proceso de selección de estudios primarios.

BD	Fase 1	Fase 2	Fase 3
Google Scholar	105	95	82
IEEE Xplore	44	41	41
		Total	***123***

- Intervención: Técnicas o Estrategias de prueba del Software.
- Comparación: Calidad del Software.
- Resultados: Mejoramiento del desempeño y rendimiento del software.
- Contexto: Industria o Academia.

Con el análisis PICOC y las preguntas de investigación formuladas, se identificaron posibles categorías que permitiesen al investigador clasificar la información resultante del análisis de los estudios primarios que resulten seleccionados (ver Tabla 1).

3.2. Bases de datos

De acuerdo con el objetivo de la investigación propuesta, para el Mapeo Sistemático se seleccionaron las siguientes Bases de Datos (BD):

- Google Scholar: una BD de artículos académicos (de una amplia gama de fuentes, pero principalmente artículos de revistas, actas de congresos, informes técnicos y disertaciones) sin restricciones de idioma, revistas o geográficas, lo cual permite acceder a literatura que no está disponible en otras BD. Si bien dicha base de datos no asegura la disponibilidad de los textos de los artículos, en el caso de nuestro estudio, es suficiente el acceso a los resúmenes de los mismos.
- IEEE *Xplore*: una BD de investigación académica en las áreas de Ciencias de la Computación, Ingeniería Eléctrica y Electrónica. Si bien los artículos de esta BD son de acceso restringido, los resúmenes son de libre acceso.

3.3. Cadena de búsqueda

Con base en los resultados obtenidos con la metodología PICOC, se elaboró la siguiente cadena genérica de búsqueda para realizar el rastreo:

("software testing" AND "software development") AND "testing software"

La cadena fue configurada en función de los repositorios considerados; las cadenas de búsqueda resultantes permitieron realizar un rastreo en los repositorios de *Google Scholar* e *IEEE Xplore* para obtener información de los últimos 10 años con respecto a las pruebas de software de manera más precisa; dichas cadenas fueron:

Google Scholar: *allintitle: Software AND Testing AND (Types OR Techniques OR Strategies).*

IEEE Xplore: *("Document Title":Software Testing) AND (("Abstract":Techniques) OR ("Abstract":Types) OR ("Abstract":Strategies)).*

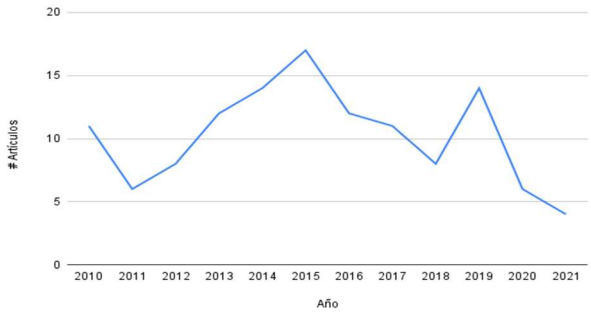


Fig. 1. Frecuencia de estudios seleccionados por año de publicación.

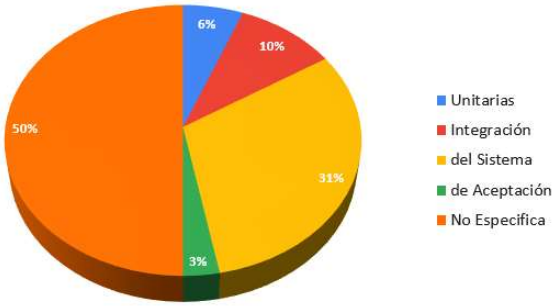


Fig. 2. Número de estudios primarios seleccionados de acuerdo con el criterio de nivel de prueba.

3.4. Criterios de inclusión y exclusión

Para filtrar la búsqueda y reducir los resultados únicamente a aquellos documentos relevantes para el objeto de investigación planteado, el cual permitiese dar respuesta a las preguntas de investigación, se definieron y utilizaron algunos criterios de inclusión y exclusión; estos criterios permitieron obtener una mejor organización de los resultados obtenidos en las búsquedas realizadas en este estudio.

Criterios de Inclusión:

- Publicaciones entre 2010 y 2021.
- Artículos publicados en revistas.
- Artículos con títulos y resúmenes en idioma inglés.

Criterios de Exclusión:

- Artículos de difusión.
- Artículos duplicados.

Tabla 3. Estudios primarios seleccionados de acuerdo al tipo de prueba reportado.

Nombre	#
Adaptativas	1
de Automatización	13
de Concurrencia	1
de Regresión	5
Escalabilidad	1
Estrés	1
Mantenibilidad	2
No Especifica	14
Rendimiento	73
Seguridad	6
Usabilidad	5

4. Ejecución del estudio

El estudio reportado fue realizado durante el mes de julio de 2021, en dicho estudio se filtraron los resultados recuperados de las búsquedas en los repositorios de *Google Scholar* e *IEEE Xplore* para excluir aquellos resultados que no estén alineados con los objetivos del mapeo sistemático de acuerdo con las tres fases definidas para el estudio:

- Fase 1: Aplicación de la cadena de búsqueda en las BD seleccionadas.
- Fase 2: Aplicación de los criterios de inclusión al conjunto de estudios obtenidos en la fase 1.
- Fase 3: Aplicación de los criterios de exclusión al conjunto de estudios obtenidos en la fase 2.

La tabla 2 presenta el número de estudios obtenidos al final de cada fase del proceso de selección.

Durante las fases anteriormente mencionadas, se descartaron un total de 26 artículos de ambas bases de datos, esto porque algunos no estaban relacionados con el tema de pruebas de software, no se apegaban a los criterios mencionados o estaban en otro idioma desde su descripción. Así pues, se obtuvo un total de 123 artículos reportados que están directamente relacionados con las pruebas de software en la última década, esto incluyendo investigaciones sobre su importancia e innovación en el área, así como experimentos y estudios de caso tanto en el ámbito industrial como en el académico. De cada estudio seleccionado se extrajo información con base en el esquema de clasificación predefinido, analizando de manera minuciosa el resumen de cada uno de los artículos seleccionados, lo anterior, en virtud del objetivo y tiempo disponible para el estudio.

5. Resultados

A continuación, se plantean los resultados obtenidos del análisis de los 123 estudios primarios seleccionados de las dos bases de datos, 82 de Google Scholar [13]-[94] y 41 de IEEE Xplore [95]-[135]. Resulta importante mencionar que se observó que buena parte de estudios analizados no proporcionaban información de manera tan detallada como para clasificarlos en una de las categorías identificadas, por lo que en algunos casos como los niveles y estrategias de prueba, se incluyó la categoría de "No Especifica".

PI01 ¿Cuál es la distribución en la última década de los estudios primarios publicados sobre pruebas del software?

La figura 1 ilustra el número de publicaciones realizadas por año en la última década; como puede observarse, en la primera mitad de la década hubo una tendencia creciente de publicaciones, no obstante —con excepción de 2019— en los últimos cinco años, la tendencia cambio a decreciente, aunque es claro que 2020 y 2021 han sido años atípicos en todas las actividades del ser humano —y la investigación no ha ido la excepción— debido a la pandemia del COVID-19.

PI02 ¿Cuáles son los niveles de prueba del software abordados por los estudios primarios en la última década?

De acuerdo con las categorías consideradas (Unitarias, Integración, del Sistema, de Aceptación), se obtuvo un total de treinta y ocho artículos que se orientaban a la investigación de pruebas del sistema, mientras que únicamente doce se orientaban a pruebas de integración, siete a unitarias y cuatro al nivel de pruebas de aceptación (ver figura 2). Resulta importante destacar la mayoría de los estudios analizados (62) fueron ubicados en la categoría de "No Especifica" dado a que en el resumen del artículo no se proporcionaba la clasificación de manera específica.

PI03 ¿Qué tipo de pruebas de software han sido las más reportadas en la última década?

Como puede observarse en la tabla 3, a pesar de que la gran mayoría de los artículos reportados fueron clasificados como pruebas de software que se orientan al rendimiento del mismo, existen otros que están clasificados dentro de la categoría de pruebas automatizadas que en su mayoría son mejoras o propuestas nuevas; por otro lado, existe un subconjunto de estudios primarios reportan otro tipo de pruebas, aunque en menor número.

PI04 ¿Cuáles son los factores de calidad que se han evaluado en el software durante la última década?

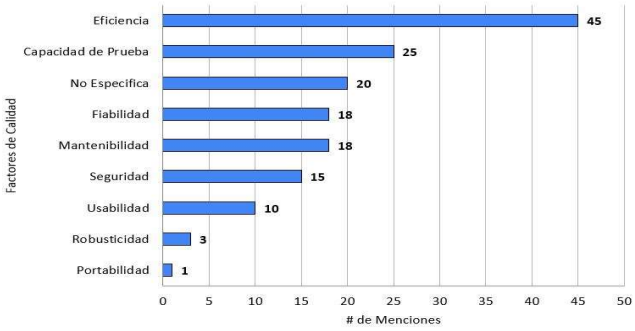


Fig. 3. Estudios primarios seleccionados de acuerdo con el factor de calidad abordado.

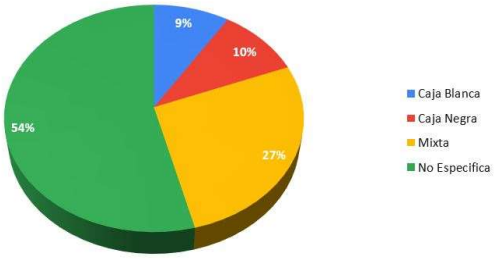


Fig. 4. Estudios primarios seleccionados de acuerdo con la estrategia de prueba seleccionada.

En relación con los factores o criterios de calidad identificados, la figura 3 ilustra la distribución por factor de calidad identificado en el estudio, por cada uno del ciento veinte y tres. Es importante mencionar que en al menos diez artículos únicamente se seleccionó un factor de calidad, mientras que en el resto era común ver más de un criterio de calidad y combinar algunos como la seguridad, capacidad de prueba, robusticidad y eficiencia. Así pues, los factores con mayor número de menciones, como puede observarse en la figura, fueron la eficiencia y la capacidad de prueba, mientras que la portabilidad y robusticidad fueron factores poco evaluados a lo largo de los artículos analizados.

Cabe destacar que el aspecto de la calidad en las pruebas denota la importancia y preocupación que existe con respecto a los aspectos de software que más se ven afectados por la falta de pruebas, investigación e innovación en el campo, así pues, como se observa en la figura, la mayoría de los estudios caen dentro de aspectos como la eficiencia, capacidad de prueba, fiabilidad y mantenibilidad; estas categorías podrían ser consideradas como primordiales para asegurar la calidad del software, lo cual confirma lo que anteriormente se pensaba con respecto a la preocupación que existe por asegurar dicha calidad en los productos en la última década.

PI05 ¿Cuáles son las características de las estrategias utilizadas en las pruebas de software que han sido reportadas en la última década?

Con el análisis de los estudios secundarios, se obtuvo que treinta y tres de los resultados aplicaban una estrategia de prueba mixta que incluía el uso de estrategias de

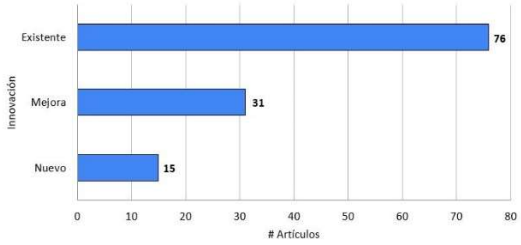


Fig. 5. Número de estudios primarios seleccionados de acuerdo con el nivel de innovación documentado.

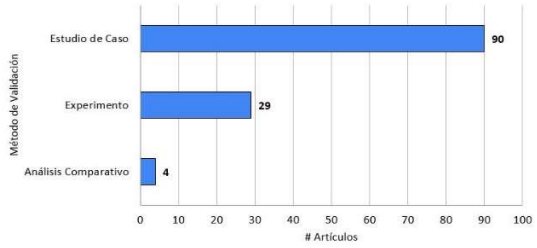


Fig. 6. Frecuencia en los Métodos de validación utilizados por los estudios primarios seleccionados.

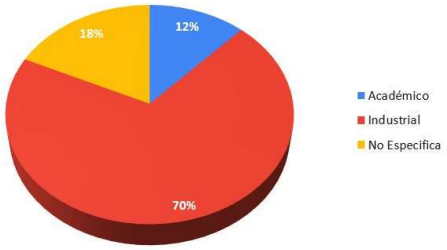


Fig. 7. Estudios primarios en función del contexto del estudio.

caja blanca y de caja negra, mientras que doce resultados reportaron aplicar estrategias de caja negra y once de caja blanca (Véase la figura 4).

Cabe destacar que sesenta y siete artículos no lograron clasificarse dado que no se mencionaba de manera explícita el tipo de estrategia de prueba seleccionado.

Por su parte, en relación con el nivel de innovación de la técnica, la figura 5 ilustra los resultados del análisis a los estudios seleccionados; se pudo contabilizar que setenta y seis de los artículos se enfocaron en técnicas o metodologías ya existentes, treinta y uno en mejoras y quince en nuevas propuestas para el área de pruebas.

Únicamente uno de los artículos no logró ser clasificado, sin embargo, se pudo observar que los pocos artículos con técnicas "nuevas" se comenzaron a volver más constantes a partir del 2015.

PI06 ¿Cuáles han sido los métodos de validación utilizados en la última década por los estudios primarios vinculados con el área de pruebas del software?

De análisis a los ciento veintitrés estudios seleccionados se pudo contabilizar los dos métodos de validación recurridos son los estudios de caso (73%) y los experimentos (29%), dos de los métodos de investigación empírica más citados en la literatura [136].

De igual forma, se hallaron cuatro artículos considerados análisis comparativos, el cual es un método de investigación que permite, como su nombre lo dice, una técnica o metodología con otra. Así pues, la figura 6 ilustra la información antes reportada.

PI07 ¿En qué contexto han sido desarrollados los estudios primarios vinculados con el área de pruebas del software durante la última década?

En relación con el contexto en el que se realizaron los estudios primarios seleccionados, se pudo observar que la mayoría se desarrolla en el ámbito industrial (86 artículos), no obstante, un alto porcentaje de éstos hacía hincapié en la investigación con respecto al ámbito de pruebas sin importar el contexto como tal (No Especificado: 22) y solo quince trabajos se reportan en el ámbito académico (ver figura 7).

Finalmente, es importante también mencionar que se obtuvieron pocos estudios de aplicación de pruebas de software en sistemas reales o experimentos aplicados al área industrial en software latente, lo cual es un indicador de que aún en la última década todavía no se profundiza en la investigación de pruebas de software en general, sobre todo para sistemas que requieren niveles altos de calidad o seguridad.

6. Conclusiones y trabajos futuros

Los Estudios de Mapeo son estudios con un alcance amplio, dado que su principal objetivo es presentar una visión global sobre un tema de interés e identificar la cantidad y tipo de investigación y resultados disponibles sobre el mismo. Con el estudio realizado en el ámbito de la Ingeniería de Software, en particular sobre el área de Pruebas de Software, se pudo observar que existe un bajo porcentaje de investigación en la que se proponen técnicas nuevas, sin embargo, la investigación se ha mantenido constante —aunque con altibajos— a lo largo de la última década.

Del análisis de los estudios primarios seleccionados se pudo identificar que las técnicas de tipo funcional son las más analizadas y que los aspectos de eficiencia y capacidad de prueba, son los criterios de calidad a los que se le ha prestado mayor atención. En cuanto a las estrategias de prueba, nos llamó la atención el uso de esquemas mixtos, es decir, una combinación de pruebas de caja blanca y de caja negra. Por otro lado, un aspecto que despertó nuestra curiosidad fue que la mayoría de los tipos de prueba fueron dirigidas al rendimiento del producto, pero también se centraban en técnicas no tan usadas que son importantes como las pruebas automatizadas.

En cuanto a los métodos de investigación empírica utilizados, pudimos observar que los estudios de caso, así como los experimentos, son los más recurridos. Finalmente, en cuanto al contexto en el que se realizaron los estudios, en esta muestra de 123 trabajos seleccionados, el industrial fue en el que se ubicaron la mayoría de los estudios analizados.

Otro de los hallazgos interesantes, es que se identificaron algunos estudios relacionados con dominios que comúnmente no se mencionan en la gran mayoría de los estudios sobre pruebas de software, nos referimos a estudios en el ámbito de

pruebas en sistemas embebidos, pruebas de seguridad en sistemas de aviación o en microprocesadores y pruebas en aplicaciones móviles; dichos artículos llamarón la atención de los autores ya que, aunque no todos buscaban innovar en alguna técnica o estrategia, analizaban temas que no son comunes en el área de pruebas y por tanto pudieran ser nichos de oportunidad para la investigación en el área de las pruebas del software.

Resulta importante destacar también, que en la actualidad, los sistemas software comienzan a abordar problemáticas diferentes a las que fueron abordadas en el siglo pasado, las soluciones comienzan a incorporar técnicas de inteligencia artificial, abordan grandes volúmenes de datos, y requieren de mejores esquemas de seguridad; por tanto, requieren de técnicas novedosas de prueba para evaluar y asegurar la calidad de los mismos.

Finalmente, los autores consideran que en el ámbito de las pruebas de software, existen problemáticas que pueden abordarse por primera vez en un futuro mediato, por lo que resulta un área de oportunidad para continuar con un estudio primario, como es estudio tipo encuesta propuesto en [137] en torno a evaluar la madurez de las prácticas de prueba; o en su caso, con un estudio secundario de mayor especificidad, como podría ser una revisión sistemática de literatura sobre las características de las pruebas de software que incorporan técnicas de Inteligencia Artificial.

Referencias

1. Abran, A., Moore J.: SWEBOK - Guide to the software engineering body of knowledge. IEEE CS Professional Practices Committee (2004)
2. Bourque, P. Fairley, R.: Guide to the software engineering body of knowledge. IEEE Computer Society Press (2014)
3. Boehm, B.: Software engineering economics. Prentice Hall (1981)
4. Burnstein, I.: Practical software testing: A process-oriented approach. Springer (2003)
5. Hetzel, B: A complete guide to software testing, QED Information Sciences, Inc (1988)
6. IEEE Standard glossary of software engineering terminology. Standard Coordinating Committee of the Computer Society. IEEE (1990)
7. Pressman, R., Maxim, B.: Software engineering: A practitioner's approach. 8th edn. McGraw-Hill Publishing (2015)
8. Sánchez Alonso, S., Sicilia Urban, M. Á., Rodriguez García, Daniel: Ingeniería del software: Un enfoque desde la perspectiva del SWEBOK. Alfaomega (2012)
9. Laporte, C., April, A.: Software quality models. Software Quality Assurance, 1st edn (2018)
10. Genero, M. Cruz-Lemus, J. Piattini, M.: Métodos de investigación en ingeniería de software (2014)
11. Petersen, K. Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: Proceedings 12th International Conference on Evaluation and Assessment in Software Engineering (2008) doi: 10.14236/ewic/EASE2008.8
12. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Evidence Based Software Engineering (2007)
13. Adamsen, C.: Automated testing techniques for event-driven and dynamically typed software applications. Doctoral Thesis, Aarhus University (2018)
14. Akour, M., Falah, B., A., Bouriat, S., Alemerien, K.: Mobile software testing: Thoughts, strategies, challenges, and experimental study. International Journal of Advanced Computer Science and Applications, vol. 7, no. 6, pp. 12–19 (2016) doi: 10.14569/ijacsa.2016.070602

15. Alharthi, A. S.: Software Testing of Mobile Applications: Techniques and Challenges. SDIWC Organization (2014)
16. Sadia, A., Yaser, H., Shariq, H., Shunkun, Y.: Enhanced regression testing technique for agile software development and continuous integration strategies. Software Quality Journal, vol. 28, pp. 1–27 (2020)
17. Alkawaz, M. H., Silvarajoo, A.: A survey on test case prioritization and optimization techniques in software regression testing. In: IEEE 7th Conference on Systems, Process and Control (ICSPC). IEEE, pp. 59–64 (2019)
18. Alnafjan, K., Hussain, T., Ullah, H., Paracha, Z.: Comparative analysis and evaluation of software vulnerabilities testing techniques. International Journal of Computer and Information Engineering, vol. 7, no. 6, pp. 687–692 (2013)
19. Alnafjan, K., Hussain, T., Khan, G.F., Ullah, H., Alghamdi, A.S.: Using AHP to compare and evaluate software securuty testing techniques. Acta Press.
20. Alsaedi, O.: Improving student skills on software testing techniques and team coordination using a zero-fidelity collaborative simulation game. Tesis Doctoral, (2019) doi: 10.2316/P.2013.796-022
21. Alzubaidy, L., Laheeb, M., Alharid, B.: Proposed software testing using intelligent techniques. Intelligent Water Drop and Ant Colony Optimization Algorithm (2013)
22. Bagchi, T.: Statistical models for software defects and testing strategies. Association for Computing Machinery, vol. 34, no. 2 (2009) doi: 10.1145/1507 195.1507202
23. Omri, F. N.: Weighted statistical testing based on active learning and formal verification techniques for software reliability assessment. Thesis Doctoral in Karlsruher Instituts für Technologie (2018)
24. Bhargava, D., Veda, A.: A different techniques and strategies for software testing. International Journal of Engineering & research technology, vol. 2, no. 12 (2013)
25. Fleischer, C., Sauer, D. U., Barreras, J. V., Schaltz, E., Christensen, A. E.: Development of software and strategies for battery management system testing on HIL simulator. In: 11th International Conference on Ecological Vehicles and Renewable Energies, pp.1–12 (2016)
26. Chaudhary, S.: Latest software testing tools and techniques: A review. International Journal, vol. 7, no. 5 (2017)
27. Chauhan, R. K., Singh, I.: Latest research and development on software testing techniques and tools. International Journal of Current Engineering and Technology, vol. 4, no 4, pp. 2368–2372 (2014)
28. Chen, Z.: Program inspection and testing techniques for code clones and refactorings in evolving software. Tesis Doctoral, University of Nebraska at Omaha (2017)
29. Cotroneo, D., Pietrantuono, R., Russo, S.: Testing techniques selection based on ODC fault types and software metrics. Journal of Systems and Software, vol. 86, no 6, pp.1613–1637 (2013)
30. Gautam, S., Nagpal, B.: Descriptive study of software testing & testing tolos. International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 6 (2016)
31. Dams, G. & Rajadoral, K.: Evaluating testing strategies and processes in software development and porting software applications. International Journal of Software Engineering, vol.9, no. 2 (2016)
32. de Olivera, I., de Souza, S.: Study and definition of project attributes for selection of testing techniques for concurrent software. In: Anais Estendidos do X Congresso Brasileiro de Software: Teoria e Prática, SBC, pp. 24–30 (2019)
33. Deak, A., Stalhane, T., Sindre, G.: Challenges and strategies for motivating software testing personnel. Information and software Technology, vol. 73, pp. 1–15 (2016)
34. Deligiannis, P.: Scalable techniques for analysing and testing asynchronous software systems. Thesis of Imperial College London (2017)

35. Dhiman, R.: Design and implemetartion of software testing techniques in cloud generating effective test cases for element based systems. Journal on Recent Innovation in Cloud Computing, Virtualization and Web Applications, vol. 2, no. 2 (2018)
36. Du, S. Y., Wang, J. S., Chen, Z. W., Ai, D. M.: Testing techniques of the software network interface based on the capture and analysis of the packet. Applied Mechanics and Materials. Trans Tech Publications, pp. 1786–1791 (2013)
37. Farooq, S. U., Quadri, S.: Empirical evaluation of software testing techniques–need, issues and mitigation. Software Engineering: An international Journal, vol. 3, no. 3, pp. 41–51 (2013)
38. Fefei, M.: Constraint solving techniques for software testing and analysis (2010) doi: 10.1145/1810295.1810407
39. Gómez, O. S., Cortés Verdín, M. K., Pardo, C.: Efficiency of software testing techniques: A controlled experiment replication and network meta-analysis. e-Informatica Software Engineering Journal, vol. 11, no. 1 (2017)
40. Hooda, I., Chhillar, R.: Software test process, testing types and techniques. International Journal of Computer Applications, vol. 111, no. 3 (2015) doi: 10.5120/19597-1433
41. Hussain, T., Singh, S., Motilal, S.: A comparative study of software testing techniques viz. white box testing black box testing and grey box testing, pp. 2350–1294 (2015)
42. Santos, I. Souza, S. Melo, S.: Extended Abstract - CTDSI/CTCCSI 2021 - Study and definition of project attributes for selection of testing techniques for concurrent software. In: Anais Estendidos do XVII Simpósio Brasileiro de Sistemas de Informação, pp. 103–105 (2021)
43. Isha, S.: Software testing techniques and strategieg. Department of Computer Science. SBMNE College (2014)
44. Jain, M., Gopalani, D.: Aspect oriented programming and types of software testing. In: Second International Conference on Computational Intelligence & Communication Technology (CICT). IEEE, pp. 64–69 (2016)
45. Jan, S., Shah Ullah, S. T., Johar, Z., Shah, Y., Khan, F.: An innovative approach to investigate various software testing techniques and strategies. International Journal of Scientific Research in Science, Engineering and Technology, vol. 2, no. 2, pp. 682–689 (2016)
46. Anil Job, M.: Automating and optimizing software testing using artificial intelligence techniques. International Journal of Advanced Computer Science and Applications, vol. 12, no. 5 (2021)
47. Kapur, P. K., Singh, O., Shrivastava, A., Singh, Jyotish, N. P.: A software up-gradation model with testing effort and two types of imperfect debugging. In: Proceedings of International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, pp. 613–618 (2015)
48. Kapur, P. K. Singh, O., Shrivastava, A.: Optimal price and testing time of a software under warranty and two types of imperfect debugging. Journal of System Assurance Engineering and Management, vol. 5, no. 2, pp. 120–126 (2014)
49. Jovanovic, I.: Software testing methods and techniques, The IPSI BgD Transactions on Internet Research, pp. 30 (2009)
50. Kolay, P., Simha, P. V: Entry and sustainable growth strategies for firms offering independent software testing services (2010)
51. Kooli, M., Kaddachi, F., DiNatale, G., Bosio, A., Benoit, P., Torres, L.: Computing reliability: On the differences between software testing and software fault injection techniques. Microprocessors and Microsystems, vol. 50, pp. 102–112 (2017)
52. Sawant, A., Bari, P., Chawan, P.: Software testing techniques and strategies. International Journal of Engineering Research and Applications(IJERA), vol. 2, no. 2, pp. 113–117 (2016)

53. Sharma, M., Kumar, V., Kumari, M., Rd, Y.: Text data generation technique for object oriented software with comparison among black box testing and white box testing techniques (2016)
54. Lazic, L.: Application example of triz and taguchi"s robust design techniques to software testing. vol 10 (2018)
55. Lopez-Herrejon, R. E., Ferrer, J., Chicano, F., Egyed, A., Alba, E.: Comparative analysis of classical multi-objective evolutionary algorithms and seeding strategies for pairwise testing of software product lines. In: IEEE Congress on Evolutionary Computation (CEC), IEEE, pp. 387–396 (2014) doi: 10.1109/CEC.2014.6900473
56. Feifei, M.: Constraint solving techniques for software testing and analysis. In: International Conference on Software Engineering. IEEE, pp. 417–420 (2010)
57. Madhu, B., Jigalur, M., Lokesha, V.: A study on agile software testing: Emergence and techniques. African Journal of Mathematics and Computer Science Research, vol. 3, no. 11, pp. 288–289 (2010)
58. Malik, S.: Software testing: Essential phase of SDLC and a comparative study of software testing techniques. International Journal of System and Software Engineering, vol. 5, no. 2, pp. 38–45 (2017)
59. Manikandan, L. C., Selvakumar, R. K.: A Study on software process framework and testing techniques. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 94–101 (2019)
60. Melton, R.: System level integration and test leveraging software unit testing techniques. In: 29th Aerospace Testing Seminar (2015)
61. Mwambe, O.: Selection and application of software testing techniques to specific conditions of software projects. International Journal of Computer Applications, vol. 975, pp. 8887 (2013)
62. Nehra, E.: A review paper on software testing techniques and tools. ZENITH International Journal of Multidisciplinary Research, vol. 9, no. 5, pp. 129–139 (2019)
63. Oliveira Neto, F. G., Torkar, R., Machado, P. D.: An initiative to improve reproducibility and empirical evaluation of software testing techniques. In: 37th International Conference on Software Engineering - New Ideas and Emerging Results, IEEE, pp. 575–578 (2015) doi: 10.1109/ICSE.2015.197
64. Ortiz, F.: Scientific test and analysis techniques for software testing. Scientific Test and Analysis Techniques Center of Excellence (STAT COE), vol. 17 (2015)
65. Patidar, R., Sharma, A. Dave, R.: Survey on manual and automation testing strategies and tools for a software application. International Journal of Advanced Research in Computer Science and Software Engineering, vol. 7, no. 44 (2017)
66. Patil, M.: The overview of software testing: types, methods, and levels. Journal of Software Engineering Tools & Technology Trends, vol. 8, no. 1, pp. 11–14 (2021)
67. Pitchford, M.: Embedded software quality, integration, and testing techniques. Software Engineering for Embedded Systems. Newnes. pp. 269–338 (2019)
68. Poonam, P. D.: Software testing strategies and methodologies. International Journal of Advanced Research Trends in Engineering and Technology, vol. 3, no. 4 (2016)
69. Razia, A., Uma, K., Sairamesh, L. Kannan, A.: Improvement on software testing techniques and tools. Journal of Advanced Research in Dynamical and Control Systems, vol. 9, no. 2, pp. 1049–1058 (2017)
70. Reddy G. V., Chandrasekhar, A.: Tools and techniques for testing of flight critical software. Defence science journal, vol. 49, no. 4, pp. 317–322 (2013)
71. Rexhepi B., Rexhepi, A.: Software testing techniques and principles. Knowledge International Journal, vol. 28, no. 4, pp. 1383–1387 (2018)
72. Sancheti, V., Sharma, G. S.: An initiative to improve quality of software testing techniques and calculating total number of failures using bayesian method. International Journal of New Technology and Research, vol. 4, no. 6, pp. 95–97 (2018)

73. Santos, I.: Study and definition of project attributes for selection of testing techniques for concurrent software. Tesis Doctoral, Universidade de São Paulo (2019)
74. Sathyavathy, V.: Software testing techniques for embedded systems and applications. International Journal of Scientific Research in Computer Science Applications and Management Studies, vol. 7, no. 3 (2018)
75. Schneidewind, N.: Software testing and reliability strategies. Journal of Aerospace Computing, Information, and Communication, vol. 7, no. 9, pp. 294–307 (2010)
76. Selvaprita, P. B.: Different software testing strategies and techniques. International Journal of Science and Modern Engineering, vol. 2, no. 1, pp. 2319–6386 (2013)
77. Sharma, S., Magow, R., Kathpal, S., Jatain, A.: Software testing techniques and experimental research drawn from inferences. International Journal of Advanced Engineering and Global Technology, vol. 2, no. 1 (2014)
78. Shuaibu, I., Machina, M., Muazzamu, I.: Investigation onto the software testing techniques and tools: An evaluation and comparative analysis. International Journal of Computer Applications, vol. 177, no. 23, pp. 8887 (2019)
79. Sneha, K., Malle, G. M.: Research on software testing techniques and software automation testing tools. In: International Conference on Energy, Communication, Data Analytics and Soft Computing. IEEE, pp. 77–81 (2017)
80. Subashini, B., Sundaravadivazhagan, B., Ashik, M.: Amalgamation and characterization for test Suite enhancement in software testing using data mining techniques. Materials Today: Proceedings (2021)
81. Subhiyakto E. R., Utomo, D. W.: Software testing techniques and strategies use in novice software teams. SISFO, vol 5, no. 5 (2016)
82. Saha T., Palit, R.: Practices of software testing techniques and tools in Bangladesh software industry. In: IEEE Asia-Pacific Conference on Computer Science and Data Engineering, pp. 1–10 (2019)
83. Taley M., Pathak, B.: Comprehensive study of software testing techniques and strategies: a review international journal of engineering and technical research, vol. 9, no. 8 (2020)
84. Thakare, S., Chavan, S., Chawan, P. M.: Software testing strategies and techniques. International Journal of Emerging Technology and Advanced Engineering, vol. 2, pp. 980–986 (2012)
85. Umar, M. A.: A study of software testing: categories, levels, techniques, and types. TechRxiv. Preprint (2020) doi: 10.36227/techrxiv.12578714.v1
86. Umar, M. A.: Comprehensive study of software testing: Categories, levels, techniques, and types. International Journal of Advance Research, Ideas and Innovations in Technology, vol. 5, no. 6, pp. 32–40 (2019)
87. Viksne, K.: Desktop software testing techniques and tools. Thesis of Institute of Information Technology (2011)
88. Saavnet, K. V., Jasuna, V.: The pragmatic review on code coverage and analysis techniques in software testing. International Journal of Computing and Corporate Research, vol. 5, no. 3 (2015)
89. Vos, T., Marin, B., Escalona, M. J., Marchetto, A.: A methodological framework for evaluating software testing techniques and tools. In: 2th international conference on quality software, IEEE, pp. 230–239 (2012)
90. Vos, T., Marín, B., Panach, I., Baars, A., Ayala, C., Franch, X.: Evaluating software testing techniques and tolos. In: Proceedings of Jornadas de Ingeniería del Software y Bases de Datos, pp. 531–536 (2011)
91. Xu, L.: The research of software testing techniques of software major and applied curriculum standards. Liaoning Higher Vocational Technical Institute Journal, vol. 9 (2011)
92. Yu, N. T.: Utilization and deployment of software testing tools and techniques. International Journal of Trend in Research and Development, vol. 6, no. 1 (2019)

93. Zhang, H. C.: Research on new techniques and development trend of software testing. Advanced materials research, Trans Tech Publications Ltd, pp. 1298–1301 (2013)
94. Żukowicz, M.: Software testing: rationale for teaching and creating test strategies using the conclusions drawn from the "No free lunch" Thory. General and Professional Education, vol. 3, pp. 61–70 (2015)
95. Arcuri, A.: A theoretical and empirical analysis of the role of test sequence length in software testing for structural coverage. IEEE Transactions on Software Engineering, vol. 38, no. 3, pp. 497–519 (2012)
96. Krieg, A., Preschern, C., Grinschgl, J., Steger, C., Kreiner, C., Weiss, R., Bock, H., Haid, J.: Power and fault emulation for software verification and system stability testing in safety critical environments. IEEE Transactions on Industrial Informatics, vol. 9, no. 2, pp. 1199–1206 (2013)
97. Zachariah, B.: Analysis of software testing strategies through attained failure size. IEEE Transactions on Reliability, vol. 61, no. 2, pp. 569–579 (2012)
98. Henard, C., Papadakis, M., Perrouin, G., Klein, J., Heymans, P., Le Traon, Y.: Bypassing the combinatorial explosion: Using similarity to generate and prioritize t-wise test configurations for software product lines. IEEE Transactions on SE, vol. 40, no. 7, pp. 650–670 (2014)
99. Hayden, C. M., Smith, E. K., Hardisty, E. A., Hicks, M., Foster, J. S.: Evaluating dynamic software update safety using systematic testing. IEEE Transactions on Software Engineering, vol. 38, no. 6, pp. 1340–1354 (2012)
100. Martinez, D., Celeita, D., Clavijo, D., Ramos, G.: Hardware and Software Integration as a Realist SCADA Environment to Test Protective Relaying Control. IEEE Transactions on Industry Applications, vol. 54, no. 2, pp. 1208–1217 (2018)
101. Cotroneo, D., Pietrantuono R., Russo, S.: RELAI Testing: A technique to assess and improve software reliability. IEEE Transactions on Software Engineering, vol. 42, no. 5, pp. 452–475 (2016)
102. Sabena, D., Reorda M. S., Sterpone, L.: On the automatic generation of optimized software-based self-test programs for VLIW processors. IEEE Trans. on Very Large Scale Integration Systems, vol. 22, no. 4, pp. 813–823 (2014)
103. Xu, D., Xu, W., Kent, M., Thomas, L., Wang, L.: An automated test generation technique for software quality assurance. IEEE Transactions on Reliability, vol. 64, no. 1, pp. 247–268 (2015)
104. Barr, E. T., Harman, M., McMinn, P., Shahbaz, M., Yoo, S.: The oracle problem in software testing: A survey. IEEE Transactions on Software Engineering, vol. 41, no. 5, pp. 507–525 (2015)
105. Tang, E., Zhang, X., Muller, N. T., Chen, Z., Li, X.: Software numerical instability detection and diagnosis by combining stochastic and infinite-precision testing. IEEE Transactions on Software Engineering, vol. 43, no. 10, pp. 975–994 (2017)
106. Uzuncaova, E., Khurshid, S., Batory, D.: Incremental test generation for software product lines. IEEE Transactions on Software Engineering, vol. 36, no. 3, pp. 309–322 (2010)
107. Bianchi, F. A., Margara, A., Pezzè, M.: A survey of recent trends in testing concurrent software systems. IEEE Transactions on Software Engineering, vol. 44, no. 8, pp. 747–783 (2018)
108. Theodorou, G., Kranitis, N., Paschalis, A., Gizopoulos, D.: Software-Based self-test for small caches in microprocessors. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 33, no. 12, pp. 1991–2004 (2014)
109. Itkonen, J., Mäntylä, V., Lassenius, C.: The role of the tester's knowledge in exploratory software testing. IEEE Transactions on Software Engineering, vol. 39, no. 5, pp. 707–724 (2013)

110. Yin, J., Cai, K.: On the asymptotic behavior of adaptive testing strategy for software reliability assessment. IEEE Transactions on Software Engineering, vol. 40, no. 4, pp. 396–412 (2014)
111. Hu, J. Cai, K. Y., Chen, T. Y.: Adaptive and random partition software testing. IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 12, pp. 1649– 1664 (2014)
112. Fiondella, L., Gokhale, S. S.: Optimal allocation of testing effort considering software architecture. IEEE Transactions on Reliability, vol. 61, no. 2, pp. 580–589 (2012)
113. Skitsas, M. A., Nicopoulos, C. A., Michael, M. K.: DaemonGuard: Enabling O/S-Orchestrated Fine-Grained Software-Based Selective-Testing in Multi-/Many-Core Microprocessors. IEEE Transactions on Computers, vol. 65, no. 5, pp. 1453–1466 (2016)
114. Böhme, M., Paul, S.: A probabilistic analysis of the efficiency of automated software testing. IEEE Transactions on Software Engineering, vol. 42, no. 4, pp. 345–360 (2016)
115. Jamro, M.: POU-Oriented unit testing of IEC 61131-3 control software. IEEE Transactions on Industrial Informatics, vol. 11, no. 5, pp. 1119–1129 (2015)
116. Khatibsyarbini, M., Isa, M. A., Jawawi, D. N., Hamed, H. N., Mohamed Suffian, M. D.: Test case prioritization using firefly algorithm for software testing. IEEE Access, vol. 7, pp. 132360–132373 (2019)
117. Kim, M., Kim Y., Kim, H.: A comparative study of software model checkers as unit testing tools: An industrial case study. IEEE Transactions on Software Engineering, vol. 37, no. 2, pp. 146–160 (2011)
118. Khan, O., Kundu, S.: Hardware/Software codesign architecture for online testing in chip multiprocessors. IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 5, pp. 714–727 (2011)
119. Bernardi, P., Cantoro, R., De Luca, S., Sanchez, E., Sansonetti, A., Squillero, G.: Software-Based Self-Test Techniques for Dual-Issue embedded processors. IEEE Transactions on Emerging Topics in Computing, vol. 8, no. 2, pp. 464–477 (2020)
120. Georgiou, P., Kavousianos, X., Cantoro, R., Reorda, M. S.: Fault-Independent Test-Generation for Software-Based Self-Testing. IEEE Transactions on Device and Materials Reliability, vol. 19, no. 2, pp. 341–349 (2019)
121. Li, Q., Li, H., Lu, M.: Incorporating S-shaped testing-effort functions into NHPP software reliability model with imperfect debugging. Journal of Systems Engineering and Electronics, vol. 26, no. 1, pp. 190–207 (2015)
122. Luo, Q., Moran, K., Zhang, L., Poshyvanyk, D.: How do static and dynamic test case prioritization techniques perform on modern software systems? An extensive study on github projects. IEEE Transactions on Software Engineering, vol. 45, no. 1, pp. 1054–1080 (2019)
123. Duffey, R. B, Fiondella, L.: Software, hardware, and procedure reliability by testing and verification: Evidence of learning trends. IEEE Transactions on Human-Machine Systems, vol. 44, no. 3, pp. 395–405 (2014)
124. Baker, R., Habli, I.: An empirical evaluation of mutation testing for improving the test quality of safety-critical software. IEEE Transactions on Software Engineering, vol. 39, no. 6, pp. 787–805 (2013)
125. Rosero, R., Gómez, O., Rodríguez, G.: Regression testing of database applications under an incremental software development setting. IEEE Access, vol. 5, pp. 18419–18428 (2017)
126. Matias, P., Barbetta, P. A., Trivedi, K. S., Filho, P. J.: Accelerated degradation tests applied to software aging experiments. IEEE Transactions on Reliability, vol. 59, no. 1, pp. 102–114 (2010)
127. Jain, R. P., Poston, R. S., Simon, J. C.: An empirical investigation of client managers' responsibilities in managing offshore outsourcing of software-testing projects. IEEE Transactions on Engineering Management, vol. 58, no. 4, pp. 743–757 (2011)

128. Shah, S., Sundmark, D., Lindström, B. Andler, S. F.: Robustness testing of embedded software systems: An industrial interview study. IEEE Access, vol. 4, pp. 1859–1871 (2016)
129. Yu, T., Huang, Z., Wang, C.: ConTesa: Directed test suite augmentation for concurrent software. IEEE Transactions on Software Engineering, vol. 46, no. 4, pp. 405–419 (2020)
130. Fragal, V. H., Simao, A., Mousavi, M. R., Turker, U. C.: Extending HSI Test Generation Method for Software Product Lines. The Computer Journal, vol. 62, no. 1, pp. 109–129 (2019)
131. Renard, Y., Lotte, F., Gibert, G., Congedo, M., Maby, E., Delannoy, V., Bertrand, O., Lécuyer, A.: OpenViBE: An open-source software platform to design, test and use brain-computer interfaces in real and virtual environments. Presence, vol. 19, no. 1, pp. 35–53 (2010)
132. Zhang, Y., Chakrabarty, K., Peng, Z., Rezine, A., Li, H., Eles, P., Jiang, J.: Software-Based Self-Testing using bounded model checking for out-of-order superscalar processors. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Syst, vol. 39, no. 3, pp. 714–727 (2020)
133. Jiang, Z. M., Hassan, A. E.: A survey on load testing of large-scale software systems. IEEE Transactions on Software Engineering, vol. 41, no. 11, pp. 1091–1118 (2015)
134. Zhou, Z. Q., Xiang, S., Chen, T. Y.: Metamorphic testing for software quality assessment: A study of search engines. IEEE Transactions on Software Engineering, vol. 42, no. 3, pp. 264–284 (2016)
135. Wang, Z., Tang, K., Yao, X.: Multi-Objective approaches to optimal testing resource allocation in modular software systems. IEEE Transactions on Reliability, vol. 59, no. 3, pp. 563–575 (2010)
136. Malhotra, R.: Empirical research in software engineering: Concepts, analysis, and applications. CRC Press (2015)
137. Garousi V., Zhi, J.: A survey of software testing practices in Canada. Journal of Systems and Software, vol. 86, pp. 1354–1376 (2013)

ISSN 1870-4069

Aprendizaje automático para la detección de la depresión en redes sociales

Alma Partida-Herrera, Geovani Peña-Ramírez,
Eduardo Vázquez-Fernández, Arturo Pérez-Cebreros

Instituto Politécnico Nacional,
ESIME Culhuacán,
México

{partidaherreraalma, geovpe, perezcebreros}@gmail.com
eduardovf@hotmail.com

Resumen. La depresión es un problema de importancia pública que ahora se prioriza en muchas agendas de atención médica con el objetivo de prevenir futuros suicidios. Esto tiene un impacto devastador no sólo por la trágica pérdida de vidas, sino también por los familiares y amigos en duelo. Las investigaciones de cada país revelan una reducción del bienestar físico y mental, por ello la propuesta presentada en este artículo pretende detectar el sentimiento de enunciados de texto mencionados en redes sociales. En particular, nosotros examinamos los tuits mediante un clasificador bayesiano y máquinas de soporte vectorial lo que nos permite dar un paso hacia adelante para identificar el estado de salud emocional.

Palabras clave: Redes sociales, depresión, aprendizaje automático.

Machine Learning for the Detection of Depression in Social Networks

Abstract. Depression is a problem of public importance that is now prioritized in many health care agendas with the goal of preventing future suicides. This has a devastating impact not only for the tragic loss of life, but also for bereaved family and friends. The investigations of each country reveal a reduction in physical and mental well-being, for this reason the proposal presented in this article aims to detect the feeling of text statements mentioned in social networks. In particular, we examine the tweets using a Bayesian classifier and support vector machines, which allows us to take a step forward to identify the state of emotional health.

Keywords: Social networks, depression, machine learning.

1. Introducción

Hoy en día, el mundo está viviendo momentos de transformaciones, la vida diaria ha tenido un giro de 360 grados en donde el protagonista ha sido una cepa viral

denominada SARS-CoV-2, que ha causado hasta el día un poco más de cuatro millones de muertes. Más allá de las consecuencias económicas, el confinamiento social suele ser una experiencia desagradable, que puede llevar a distintos estresores que generen afecciones de la salud mental [1].

Debido a que esta situación es nueva y se encuentra en plena expansión, es aún prematuro estimar las consecuencias emocionales del brote epidémico. Sin embargo, las investigaciones realizadas en [2,3] apuntan a que el miedo a lo desconocido y la incertidumbre pueden llevar a evolucionar distintas enfermedades de salud mental como: los trastornos de estrés, ansiedad, depresión, somatización y conductas que degeneran en aumento de consumo de alcohol, tabaco y otras substancias nocivas para la salud [4].

En particular, se espera que las personas con enfermedades crónicas presenten niveles más altos de síntomas psicológicos [5]. También las personas mayores se pronostican que sean psicológicamente más vulnerables que los jóvenes en esta crisis [6]. Este proyecto surge debido a la gran problemática sobre casos de suicidio en nuestro país en jóvenes [7, 8, 9] por ello se decide desarrollar una herramienta que sea capaz de alertar sobre posibles casos y que permita evitarlos.

1.1. La depresión y la inteligencia artificial

Cabe destacar que, en México, se encontró que adultos jóvenes (es decir, entre 15 y 25 años) presentan ideas suicidas y muestran mayores estados depresivos, es decir, la depresión aparece en el 67.3% de quienes han intentado suicidarse y en el 81.1% de quienes manifiestan ideas suicidas [10]. Y las personas con enfermedades mentales tienden a revelar su condición mental en las redes sociales, como una forma de alivio [11].

Sin embargo, la investigación sobre el aprovechamiento de redes sociales para comprender trastornos de la salud del comportamiento aún está en su infancia. En Katikalapudi et al. [12] analizaron patrones de actividad web de estudiantes universitarios que podrían indicar depresión. De manera similar, en Katie et al. [13] demostraron que las actualizaciones de estado en Facebook podrían revelar síntomas de episodios depresivos.

Aunque algunas diferencias han sido observadas, como que los usuarios deprimidos usan con más frecuencia pronombres en primera persona [14] así como palabras de emociones negativas e ira. Por ello, la depresión ha sido asociada al uso de marcadores lingüísticos tales como el uso elevado de pronombres de primera persona.

Muchos otros estudios del lenguaje y la depresión se han limitado a entornos clínicos, y, por lo tanto, a analizar discursos espontáneos o ensayos escritos. En esa dirección, algunas investigaciones [15, 16] propusieron metodologías innovadoras para recopilar contenidos textuales compartidos por personas diagnosticadas con depresión. Sin embargo, no hay colecciones disponibles públicamente.

Esto se debe a que a menudo el texto se extrae de sitios de redes sociales, como Twitter o Facebook que no permiten la redistribución [17]. De ahí que estos estudios previos, nos impulsen a la detección de la depresión en las redes sociales como primer paso contra el suicidio. El punto central de los estudios de la salud mental en redes sociales ha sido, tradicionalmente, llevado a cabo mediante el uso de encuestas. En

donde el número de usuarios está limitado por aquellos que puedan completar la encuesta. Por ejemplo, en Choudhury et al. [18] solicitó a los usuarios de Twitter que hicieran la Escala de Depresión del Centro de Estudios Epidemiológicos (CES-D) y compartieran su perfil al público.

Este tipo de estudios ha producido datos de alta calidad, sin embargo, está limitado en tamaño y alcance. Por ello, en esta investigación examinaremos la depresión considerando muestras obtenidas automáticamente, de grandes cantidades de datos de Twitter. El Internet ha permitido seguir la evolución del lenguaje y nos está proporcionando un medio muy accesible para que las personas expresen sus sentimientos de forma anónima.

De ahí que, nosotros hemos adaptado el método en Coppersmith et al. [15] para la construcción de este conjunto de datos en español, procederemos a identificar auto expresiones de diagnósticos de enfermedades mentales y aprovechamos estos mensajes para construir nuestro conjunto de datos.

1.2. Análisis de sentimientos

Entre las distintas plataformas de redes sociales, Twitter ha experimentado una adopción particularmente generalizada de usuarios; es una plataforma de microblogueo donde los usuarios crean tuits que se transmiten a sus seguidores o que son enviados a otro usuario.

A partir del 2016, Twitter tiene más de 313 millones de usuarios [19], y junto con este tremendo crecimiento, Twitter también ha sido objeto de muchas investigaciones de análisis de sentimientos (SA), ya que los tuits a menudo expresan la opinión de un usuario sobre un tema de interés. La palabra sentimiento se refiere a una forma de pensar (opinión) o sentir (emoción) sobre algo [20].

Originalmente, el SA apareció con la inteligencia de negocios [21, 23], pero se ha extendido a otras áreas como la política [23, 24], medicina [25], educación [26], recomendaciones [27, 28], detección de plagios [29], influencia en las noticias [30], detección del engaño [31, 32], detección de ironías [33], clasificación de cuentas [34], entre otras.

En particular, el SA es un tema destacado de investigación en el campo de la lingüística computacional. Las tareas incluyen la clasificación de la polaridad de sentimiento expresado en el texto (por ejemplo, positivo, negativo y neutral), identificación del objetivo/tema de sentimiento e identificación del sentimiento por varios aspectos de un tema.

El problema de clasificación de la polaridad de sentimiento es a menudo modelado como bidireccional (positivo/negativo) o tridireccional (positivo/negativo/neutral) [35]. Es importante resaltar que esta tarea de detección y clasificación no es sencilla, en primer lugar, debido a que los tuits son mensajes cortos en donde los indicadores de depresión suelen manifestarse de forma muy sutil.

A pesar de ello, algunos trabajos recientes han reportado resultados alentadores en la detección de usuarios que padecen de depresión, pero aún se requieren más estudios [36,37].

En nuestra investigación, modelaremos de manera bidireccional (positivo/negativo); dejando para un próximo trabajo, evaluar más intensidades del sentimiento de depresión: fuertes positivos, fuertes negativos, leves positivos, y leves

Tabla 1. Expresiones regulares para la detección de tuits.

Palabra	Expresión regular
Depresión	(depresi[a-z]+)
Deprimido	(deprimi[a-z]+)
Frases asociadas	((problema[s]|disturbio(s)) *(mental|psicologico(s)|psiquiatrico(s))) (quiero) * (morir|morir[a-z]+) (todo(s))*(dia(s)*(trist[a-z] + problema(s))

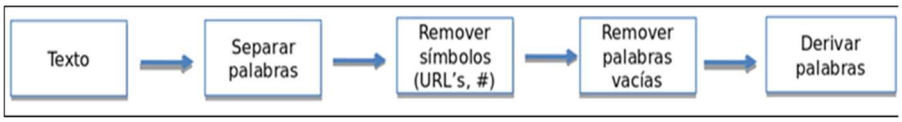


Fig. 1. Escenario del pre-procesamiento de tuits.

negativos. Detectar el sentimiento de las frases es una tarea complicada, por ejemplo: 'la vida es como el jazz, mejor si es improvisada'; el sentimiento de la opinión es positiva porque la palabra 'vida' implica algo bueno.

Sin embargo, la misma palabra en otro contexto, como se muestra en el siguiente enunciado: 'mi vida no tiene sentido', implica un sentimiento negativo – es malo porque la negación reduce lo positivo de la palabra 'vida'. Entonces el problema implica el uso del lenguaje, lo cual es un problema muy complejo y vasto.

2. Desarrollo

Para darle solución a este problema se propuso un modelo de tres fases.

2.1. Fase de recolección

Durante esta fase, se tomó ventaja de la gran cantidad de datos proporcionados por Twitter. El método de recolección se basa en dos pasos principales: primero, los tuits se filtran mediante expresiones regulares posteriormente son clasificados: en negativos y positivos.

Para adquirir los tuits para este estudio, desarrollamos una aplicación que utiliza la API de búsqueda de Twitter [38]. Para filtrar los tuits que no están escritos en español, utilizamos la biblioteca de detección de idiomas disponible gratuitamente [39]. Esta librería se basa en filtros bayesianos y tiene una precisión de 0.99 en la detección de los 53 idiomas que admite.

Los tuits se adquirieron durante 210 días (del 01 de diciembre del 2020 al 01 de julio del 2021), produciendo conjuntos de datos con aproximadamente 4000 tuits para español. Para generar el conjunto de datos de tuits con rasgos depresivos, nosotros consideramos tuits de personas que declararon haber sido diagnosticados con la enfermedad de la depresión.

En la Tabla 1, se muestran las expresiones regulares usadas para detectar a las personas que hacen referencia de la depresión en sus tuits; pero el principal objetivo es encontrar personas que hagan una declaración directa y abierta de que fueron

diagnosticados con la enfermedad de la depresión. Posteriormente, se procede a extraer los tuits de la lista de las personas que aseveraron, por medio de un tuit, tener dicha enfermedad.

2.2. Fase de preprocesamiento

El preprocesamiento de datos es un paso, a menudo, descuidado pero importante en el proceso. Implica técnicas para transformar los datos sin procesar en un formato más comprensible. Las principales son limpieza del dato, integración de datos, transformación de datos y reducción de datos.

Como podemos ver la Figura 1, nuestro mecanismo de preprocesamiento incluye:

a) Separar palabras del texto,
b) Eliminación de números y URLs que involucra un efecto sobre nuestro análisis, pero si reduce el ruido y nuestra eficiencia [40],
c) Eliminación de palabras vacías como artículos, pronombres, y preposiciones [41],
d) Derivación de las palabras, el cual se utiliza para transformar diferentes formas de palabras en una forma raíz estándar [42].

En esta fase, además de estas técnicas incorporamos un paso de ponderación mediante el algoritmo Term Frequency-Inverse Document (TF-IDF). El TF-IDF refleja la importancia de una palabra en un documento; y este nivel de importancia se incrementa cuando la palabra aparece muchas veces, al punto que podemos determinar temas de tendencia [43].

La Frecuencia de Términos (TF) es la frecuencia con la que las palabras aparecen en un documento. Para un término t_i en un documento, podemos formularlo de la siguiente manera:

$$Tf_{i,j} = n_{i,j.} \tag{1}$$

En (1), tenemos que n_{ij} es el número de ocurrencias de cada palabra t_i en el documento d_j. Por otro lado, la Frecuencia del Documento Invertido (IDF) mide la importancia general de una palabra en un documento. La podemos formular de la siguiente forma:

$$idf_i = log\ (D/df_{i,}) \tag{2}$$

En (2), tenemos que D es el número total de documentos de texto y df_i es un número de documentos el cual contiene el termino t_i por lo menos una vez. Finalmente, tenemos que TF-IDF es una combinación de TF y de IDF, la formula quedaría así:

$$Tf - idf_{i,j} = tf_{i,j} \times idf_{i.} \tag{3}$$

2.3. Fase de Identificación/Clasificación

El algoritmo de clasificación basado en máquinas de soporte vectorial (SVM) es una máquina de aprendizaje supervisado, que requiere de datos de entrenamiento y datos de prueba. Consiste en encontrar un hiperplano óptimo como la función que separa dos clases de datos.

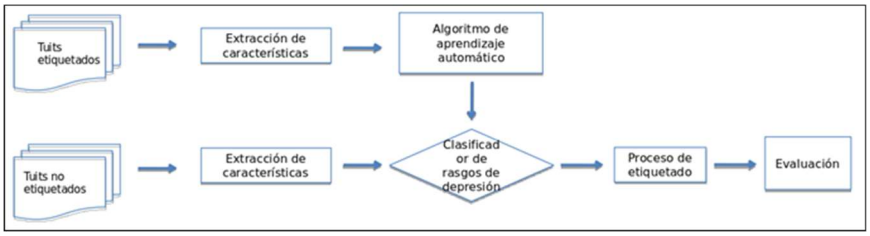


Fig. 2. Flujo del proceso de clasificación de rasgos depresivos.

La clasificación con menor error es la que se consigue con el hiperplano que maximiza el margen, esto es, cuya distancia entre el plano y los vectores soporte, sea la mayor posible. A pesar de su sencillez ha demostrado ser un algoritmo robusto y que generaliza bien en problemas de la vida real [44-48].

3. Resultados

El método propuesto, realiza una clasificación e identificación de tuits que nos permite tener una visualización precisa y directa, se puede determinar si la frase que fue extraída de Twitter tiene sentido de depresión o no y así poder ayudar a la persona que se requiera. En la Figura 2, podemos visualizar el problema de analizar los mensajes posteados en Twitter en términos de los sentimientos que estos mensajes expresan.

En donde primero nos dimos a la tarea de etiquetar un conjunto de tuits en español obtenidos bajo la metodología descrita. Adicional, cuando etiquetamos es importante considerar la presencia de la negación, debido a que la negación juega un papel muy importante en la detección de la polaridad de un mensaje (positivos se hacen negativos y viceversa).

Esta clasificación no es una tarea trivial y una de las características de Twitter es que es un tipo de comunicación informal, y con limitaciones de longitud. Esto lo hace diferente a otras investigaciones previas de análisis de sentimientos de textos convencionales. En la Tabla 2 se muestra las diez palabras con mayor frecuencia positivas y negativas respectivamente. Cabe resaltar que la palabra 'vida' aparece tanto en el lado de positivas como en el de negativas, más adelante, en la Tabla 3 explicamos el cambio de la polaridad.

Como parte de las limitaciones que tiene el presente trabajo es que, falta hacer más estudios para reducir la dispersión en donde podríamos aplicar técnicas de suavizado semántico entre otras [49].

Los resultados obtenidos por el clasificador bayesiano y las máquinas de soporte vectorial, se compararon mediante las siguientes métricas: exactitud, precisión y sensibilidad donde:

– Exactitud es una medida en porcentaje que se calcula de la siguiente forma:

$$Exactitud = (Tp + Tn) / (Tp+Tn+Fn+Fp). \quad (4)$$

– Sensibilidad positiva (5) y sensibilidad negativa (6) es el ratio de sensibilidad y es calculada de la siguiente forma:

Tabla 2. Listado de las palabras con mayor frecuencia.

Positivas	**Negativas**
vida	vida
feliz	solo(a)
mejor	mal
contento	mierda
mundo	nadie
ganar	triste
amor	llorar
trabajo	dormir
esfuerzo	sentir
positivo	tiempo

– La precisión positiva (7) y la precisión negativa (8) es el ratio de precisión y se calcula de la siguiente forma:

$$Precisión\ p = Tp/(Tp+Fp), \tag{7}$$

$$Precisión\ n = Tn/(Fn+Tn). \tag{8}$$

En la Tabla 3, se presentan cuatro tuits extraídos del conjunto de datos; podemos observar que la negación juega un papel muy importante en la detección de la polaridad de una frase (positivos se hacen negativos y viceversa), además de la negación se tiene que considerar adjetivos que acompañan al sustantivo y que cambian su cualidad.

$$Sensibilidad\ p = Tp\ /\ (Tp + Fn), \tag{5}$$

$$Sensibilidad\ n = Tn\ /\ (Fp + Tn). \tag{6}$$

4. Conclusiones

En la Tabla 4 se muestra la comparación del desempeño entre el clasificador bayesiano y las máquinas de soporte vectorial respectivamente en términos de precisión y sensibilidad. De manera similar, la Tabla 5 muestra el desempeño de los clasificadores en términos de exactitud.

La tendencia creciente de la depresión y del suicidio son un grave problema de salud pública. Sin duda, este es un problema que el sistema de salud mexicano debe enfrentar de manera urgente, por un lado, debe considerar que el país se encuentra en una etapa de incertidumbre económica (derivado de la actual pandemia), y por otro lado, que existen necesidades de atención de salud mental.

Tabla 3. Fragmentos de tuits extraídos del conjunto de datos.

Tuits	Clase
La **vida** es como el **jazz**, mejor si es **improvisada**	Positiva
La **magia** es **creer** en **ti** mismo	Positiva
La **vida** es una **mierda**	Negativa
La **vida no** tiene **sentido**	Negativa

Tabla 4. Métricas de desempeño.

Métricas	%
Sensibilidad positiva	84
Sensibilidad negativa	84
Precisión positiva	87
Precisión negativa	84

Tabla 5. Comparación de exactitud.

Métodos	%
Clasificador bayesiano	84
Máquinas de soporte vectorial	86

Nuestro método propuesto puede ser una base para más estudios de computación social y abre las puertas a futuras investigaciones sobre algoritmos de IA que hagan uso de otros datos de entrenamiento del tipo multifactorial y multinivel, como lo son las variables sociales, económicas y políticas.

Con el fin de explorar la salud mental, la idea central de la presente investigación parte del principio de clasificar un texto como positivo o negativo mediante algoritmos de IA.

Como primer paso, se describe una metodología con la que se genera un conjunto de datos en español y con ello, se establecen algunos pasos esenciales para la clasificación de rasgos depresivos.

Nosotros hemos aplicado el clasificador bayesiano y el clasificador de máquinas de soporte vectorial para la clasificación de textos con rasgos depresivos obteniendo muy buenos resultados. En futuros trabajos, procederemos a incrementar el tamaño del conjunto de datos mediante la metodología descrita, analizaremos diferentes técnicas para la representación de textos, por ejemplo, incorporaremos una reducción de dimensionalidad mediante un modelo de bolsa de palabras (BOW).

También podríamos combinar nuestros algoritmos con información de tipo multimodal que ofrece una nueva dimensión a los análisis tradicionales sobre texto, en donde podríamos tomar diferentes modalidades como datos visuales, de audio, entre otros [50-52]. Así como combinarla con otras técnicas de aprendizaje profundo mediante arquitecturas jerárquicas para incrementar la escalabilidad e incrementar la precisión de nuestro método [53, 54].

Referencias

1. McGuine, T. A., Biese, K. M., Petrovska, L., Hetzel, S. J., Reardon, C. L., Kliethermes, S., Bell, D. R., Brooks, A., Watson, A. M.: Changes in the health of adolescent athletes: A comparison of health measures collected before and during the CoVID-19 pandemic. In: Proceedings of Journal of Athletic Training (2021)
2. Wang, Q., Su, M. A.: Preliminary assessment of the impact of COVID-19 on environment – A case study of China. Science of The Total Environment, vol. 728 (2020) doi: 10.1016/j.scitotenv.2020.138915
3. Liu, C. H., Zhang, E., Wong, G. T., Hyun, S., Hahm, H. C.: Factors associated with depression, anxiety, and PTSD symptomatology during the COVID-19 pandemic: Clinical implications for U.S. young adult mental health. Psychiatry Research, vol. 290 (2020) doi: 10.1016/j.psychres.2 020.113172
4. Shigemura, J., Kurosawa, M.: Mental health impact of the COVID-19 pandemic in Japan. Psychological Trauma: Theory, Research, Practice, and Policy, vol. 12, no. 5, pp. 478–79 (2020) doi: 10.1037/tra0000803
5. Martínez-Taboas, A.: Pandemias, COVID-19 y salud mental: ¿Qué sabemos actualmente? Revista Caribeña de Psicología, vol. 4, no. 2, pp. 143–152 (2020) doi: 10.37226/rcp.v4i2.4907
6. Landry, M. D., van den Bergh, G., Hjelle, K. M., Jalovcic, D., Tuntland, H. K.: Betrayal of Trust? The Impact of the COVID-19 Global Pandemic on Older Persons. Journal of Applied Gerontology, vol. 39, no. 7, pp. 687–689 (2020) doi: 10.1177/0733464820924131
7. Rodríguez Esparza, L. J., Barraza Barraza, D., Salazar Ibarra, J., Vargas Pasaye, R. G.: Index of suicide risk in Mexico using Twitter. Journal of Social Researches, pp. 1–13 (2019) doi:10.35429/JSR.2019.15.5.1.13
8. Cabello-Rangel, H., Márquez-Caraveo, M. E., Díaz-Castro, L.: Suicide rate, depression and the human development index: An ecological study from Mexico. Frontiers in Public Health, vol. 8 (2020) doi: 10.3389/fpubh.20 20.561966
9. Cervantes, C. A., Montaño, A. M.: Estudio de la carga de la mortalidad por suicidio en México 1990-2017. Revista Brasileira de Epidemiologia, vol. 23 (2020) doi: 10.1590/1980-549720200069
10. Cañón, Buitrago, S. C., Carmona Parra, J. A.: Ideación y conductas suicidas en adolescentes y jóvenes. Revista Pediatría de Atención Primaria, vol. 20, pp. 471–489 (2018)
11. Benítez Camacho, E.: Suicidio: El impacto del COVID-19 en la salud mental. Revista de Medicina y Ética, vol. 32, no. 1, pp. 15–39 (2021) doi: 10.36105/ mye.2021v32n1.01
12. Katikalapudi, R., Chellappan, S., Montgomery, F., Wunsch, D., Lutzen, K.: Associating internet usage with depressive behavior among college students. IEEE Technology and Society Magazine, vol. 31, no. 4, pp. 73–80 (2012)
13. Katie, G., Megan Moreno, A.: Alcohol references on undergraduate males' Facebook profiles. American Journal of Men's Health, vol. 5, no. 5, pp. 413–420 (2011) doi: 10.1177/1557988310394341
14. Chung, C., Pennebaker, J.: The psychological functions of function words. Social Communication, pp. 343–359 (2007)
15. Coppersmith, G., Dredze, M., Harman, C., Hollingshead, K., Mitchell, M.: CLPsych 2015 Shared Task: Depression and PTSD on Twitter. In: Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality, pp. 31–39 (2015) doi:10.3115/v1/W15-1204

16. Martínez-Castaño, R., Pichel, J. C., Losada, D. E.: A big data platform for real time analysis of signs of depression in social media. International Journal of Environmental Research and Public Health, vol. 17, no. 13 (2020) doi: 10.33 90/ijerph17134752
17. Zivanovic, S., Martinez, J., Verplanke, J.: Capturing and mapping quality of life using twitter data. GeoJournal, vol. 85, no. 1, pp. 237–255 (2020) doi: 10.1007/s10708-018-9960-6
18. Choudhury, M., Gamon, M., Counts, S., Horvitz, E.: Predicting depression via social media. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 7, no. 1, pp. 128–1973
19. Alsaeedi, A., Zubair, M.: A study on sentiment analysis techniques of Twitter data. International Journal of Advanced Computer Science and Applications, vol. 10, no. 2, pp. 361–374 (2019)
20. Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., Díaz-Rangel, I., Suárez-Guerra, S., Treviño, A., Gordon, J.: Empirical study of machine learning based approach for opinion mining in tweets. Batyrshin, I., González Mendoza, M. (eds) Advances in Artificial Intelligence. MICAI 2012, Lecture Notes in Computer Science, vol. 7629, Springer (2013) doi: 10.1007/978-3-642-37807-2_1
21. Chaturvedi, S., Mishra, V., Mishra, N.: Sentiment analysis using machine learning for business intelligence. In: IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), pp. 2162–2166 (2017) doi: 10.1109/ICP CSI.2017.8392100
22. Garcia-Lopez, F. J., Batyrshin, I., Gelbukh, A.: Analysis of relationships between tweets and stock market trends. Journal of Intelligent & Fuzzy Systems, vol. 34, no. 5, pp. 3337–3347 (2018) doi: 10.3233/JIFS-169515
23. Bernábe Loranca, M. B., González Velázquez, E. E., Cerón Garnica, C.: Algorithm for collecting and sorting data from twitter through the use of dictionaries in Python. Computación y Sistemas, vol. 24, no. 2 (2020) doi: 10.13053/cys-24-2-3405
24. Rill, S., Reinel, D., Scheidt, J., Zicari, R. V.: PoliTwi: Early detection of emerging political topics on Twitter and the impact on concept-level sentiment analysis. Knowledge-based Systems, vol. 69, pp. 24–33 (2014)
25. Pavan Kumar, C. S., Dhinesh Babu, L. D.: Fuzzy based feature engineering architecture for sentiment analysis of medical discussion over online social networks. Journal of Intelligent & Fuzzy Systems, vol. 40, no. 6, pp. 11749–11761 (2021)
26. Gutiérrez, G., Canul-Reich, J., Ochoa Zezzatti, A., Margain, L., Ponce, J.: Mining: Students comments about teacher performance assessment using machine learning algorithms. International Journal of Combinatorial Optimization Problems and Informatics, vol. 9, no. 3, pp. 26–40 (2018)
27. Gupta, V., Singh, V. K., Mukhija, P. Ghose, U.: Aspect-based sentiment analysis of mobile reviews. Journal of Intelligent & Fuzzy Systems, vol. 36, pp. 4721–4730 (2019)
28. Wang, J., Zhang, X., Yu Zhang, H.: Hotel recommendation approach based on the online consumer reviews using interval neutrosophic linguistic numbers. Journal of Intelligent & Fuzzy Systems, vol. 34, pp. 381–394 (2018)
29. González Brito, O., Tapia Fabela, J. L., Salas Hernández, S.: Method of extraction of feature in the classification of texts for authorship attribution. International Journal of Combinatorial Optimization Problems and Informatics, vol. 12, no. 3, pp. 87–97 (2021)
30. Maldonado-Sifuentes, C. E., Sidorov, G., Kolesnikova, O.: Improved Twitter virality prediction using text and RNN-LSTM. International Journal of Combinatorial Optimization Problems and Informatics, vol. 12, no. 3, pp. 50–62 (2021)

31. Hernández Castañeda, Á., García Hernández, R. A., Ledeneva, Y., Millán Hernández, C. E.: The impact of key ideas on automatic deception detection in text. Computación y Sistemas, vol. 24, no. 3 (2020)
32. Posadas-Durán, J. P., Gómez-Adorno, H., Sidorov, G., Escobar, J. J.: Detection of fake news in a new corpus for the Spanish language. Journal of Intelligent & Fuzzy Systems, vol. 36, no. 5, pp. 4869–4876 (2019)
33. Calvo, H., Gambino, O. J., García Mendoza, C. V.: Irony detection using emotion cues. Computación y Sistemas, vol. 24, no. 3 (2020)
34. Daouadi, K. E., Rebaï, R. Z., Amous, I.: Organization, bot, or human: Towards an efficient twitter user classification. Computación y Sistemas, vol. 23, no. 2, pp. 273–279 (2019)
35. Zimbra, D., Abbasi, A., Zeng, D., Chen, H.: The state-of-the-art in Twitter sentiment analysis: A review and benchmark evaluation. ACM Transactions on Management Information Systems (TMIS), vol. 9, no. 2, pp. 1–29 (2018)
36. Zucco, C., Calabrese, B., Cannataro, M.: Sentiment analysis and affective computing for depression monitoring. In: IEEE International Conference on Bioinformatics and Biomedicine, pp. 1988–1995 (2017)
37. Vázquez-Hernández, M., Pineda, L. V., Montes-y-Gómez, M.: Identificación y pesado de términos para la detección de depresión en Twitter. Research in Computer Science, vol. 149, no. 8, pp. 465–474 (2020)
38. Trupthi, M., Pabboju, S., Narasimha, G.: Sentiment analysis on twitter using streaming API. In: IEEE 7th International Advance Computing Conference (IACC), pp. 915–919 (2017) doi: 10.1109/IACC.2017.0186
39. Balazevic, I., Braun, M., Müller, K. R.: Language detection for short text messages in social media (2016) doi: 10.48550/arXiv.1608.08515
40. Khader, M., Awajan, A. A., Al-Naymat, G.: The impact of natural language preprocessing on big data sentiment analysis. vol.16, no. 3, pp. 506–513 (2019)
41. Saif, H., Fernandez, M., He, Y., Alani, H.: On stopwords, filtering and data sparsity for sentiment analysis of Twitter. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation, pp. 810–817 (2014)
42. Jabbar, A., Iqbal, S., Tamimy, M. I., Hussain, S., Akhunzada, A.: Empirical evaluation and study of text stemming algorithms. Artificial Intelligence Review, vol. 53, no. 8, pp. 5559–5588 (2020) doi: 10.1007/s10462-020-09828-3
43. Zhu, Z., Liang, J., Li, D., Yu, H., Liu, G.: Hot topic detection based on a refined TF-IDF algorithm. IEEE Access, vol. 7, pp. 26996–27007 (2019) doi: 10.1109/ACCESS. 2019.2893980
44. Khanna, D., Sahu, R., Baths, V., Deshpande, B.: Comparative study of classification techniques (SVM, Logistic Regression and Neural Networks) to Predict the Prevalence of Heart Disease'. International Journal of Machine Learning and Computing, vol. 5, no. 5, pp. 414–419 (2015) doi: 10.7763/IJM LC.2015.V5.544
45. Lopez-Martin, C., Banitaan, S., Garcia-Floriano, A., Yanez-Marquez, C.: Support vector regression for predicting the enhancement duration of software projects. In: 16th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 562–567 (2017)
46. Toledo, G. R., Sánchez, N. A., Sidorov, G., Durán, J. P.: Identificación de cambios en el estilo de escritura literaria con aprendizaje automático. Onomázein: Revista de lingüística, filología y traducción de la Pontificia Universidad Católica de Chile, vol. 46, pp. 102–128 (2019)
47. Ramírez-García, J., Ibarra-Orozco, R. E., Cruz, A. J.: Tweets monitoring for real-time emergency events detection in smart campus. In: Mexican International Conference on Artificial Intelligence, pp. 205–213 (2020)

48. Nieto-Benitez, K., Castro-Sánchez, N. A., Jiménez-Salazar, H.: Reconocimiento de patrones para la clasificación de componentes argumentales en textos académicos en español. Research in Computing Science, vol. 149, no. 8, pp. 637-648 (2020)
49. Altinel, B., Ganiz, M. C.: Semantic text classification: A survey of past and recent advances. Information Processing & Management, vol. 54, no. 6, pp. 1129–1153 (2018)
50. Poria, S., Cambria, E., Gelbukh, A.: Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In: Conference on Empirical Methods in Natural Language Processing, pp. 2539–2544 (2015) doi: 10.18653/v1/D15-1303
51. Krishnamurthy, G., Majumder, N., Poria, S., Cambria, E.: A deep learning approach for multimodal deception detection (2018)
52. Banerjee, T., Yagnik, N., Hegde, A.: Impact of cultural-shift on multimodal sentiment analysis. Journal of Intelligent & Fuzzy Systems, pp. 5487–5496 (2021) doi: 10.3233/JIFS-189870
53. Kastrati, Z., Imran, A. S., Yayilgan, S. Y.: The impact of deep learning on document classification using semantically rich representations. Information Processing & Management, vol. 56, no. 5, pp. 1618–1632 (2019)
54. Amjad, M., Voronkov, I., Saenko, A., Gelbukh, A.: Comparison of text classification methods using deep learning neural networks. In: Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (2019)

ISSN 1870-4069

Clasificación de Diabetes Mellitus tipo II detectando factores de riesgo en un conjunto de datos

Juan Manuel Cancino-Gordillo, Mireya Tovar-Vidal

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

juan@comitan.com, mireya.tovar@correo.buap.mx

Resumen. Una de las enfermedades más importantes a nivel mundial en salud pública es la Diabetes Mellitus (DM), ya que esta es una de las enfermedades no transmisibles más severa, frecuente y con diversas complicaciones crónicas. En este documento proponemos un método para detectar los factores de riesgo más comunes en pacientes que padecen la enfermedad conocida como diabetes mellitus tipo II, a través del análisis de componentes principales. Posteriormente comprobamos los resultados utilizando estos factores como características, por medio del algoritmo J48 mejorando los resultados de la clasificación. De acuerdo a los resultados experimentales se obtiene un 86.9% de precisión, la cual es una mejora en comparación con trabajos relacionados.

Palabras clave: Aprendizaje automático, PCA, treej48, clasificación.

Classification of Type II Diabetes Mellitus Detecting Risk Factors in a Data Set

Abstract. One of the most important diseases worldwide in public health is Diabetes Mellitus (DM), since this is one of the most severe and frequent non-communicable diseases with various chronic complications. In this document we propose a method to detect the most common risk factors in patients suffering from the disease known as type II diabetes mellitus, through principal component analysis. Later we check the results using these factors as characteristics, by means of the J48 algorithm, improving the classification results. According to the experimental results, 86.9% accuracy is obtained, which is an improvement compared to related works.

Keywords: Machine learning, PCA, treej48, classification.

1. Introducción

En el campo médico el diagnóstico es la parte más crítica al momento de tratar a una persona, ya que el médico utiliza sus conocimientos para detectar ciertos patrones en el

comportamiento o estudios médicos de un paciente y llegar a una conclusión que se traduce en el tratamiento o medicación. Entre la amplia gama de enfermedades, la Diabetes Mellitus (DM) y sus variantes son considerados severos por las diversas complicaciones crónicas y requiere la atención de diferentes especialistas para el tratamiento de una persona.

Existen dos variantes de la DM, tipo I y tipo II. La más común es la Diabetes Mellitus tipo II (DMT2), la cual es una enfermedad en la que el organismo no genera suficiente insulina para procesar la glucosa en la sangre dejando mucho de este material circulando en el sistema sanguíneo. En México se atribuye el 11.8% de muertes desde el 2005 [1] y un 63% como causa de muerte principal de enfermedades crónicas a nivel mundial en el 2015 [2].

El propósito de este trabajo es la detección de factores de riesgo que pueden originar la DMT2, a partir del análisis de componentes principales en un conjunto de expedientes clínicos. Posteriormente se aplica un algoritmo de aprendizaje automático para corroborar que esos factores contribuyen en la detección de DMT2, al mejorar la precisión en los resultados experimentales.

El artículo está distribuido en cuatro secciones. En la sección 2 se presentan los trabajos relacionados con esta investigación. En la sección 3 se incluye una breve explicación a la metodología, algoritmos y métricas utilizadas. En la sección 4 se presentan los resultados experimentales y finalmente se incluyen las conclusiones del trabajo realizado.

2. Trabajos relacionados

A continuación, se describen algunos trabajos relacionados con el uso de algoritmos de aprendizaje automático.

En el trabajo presentado por AlJarullah et al. [3] se utilizan árboles de decisión con un conjunto de datos enfocado en mujeres para la detección de la diabetes. Este trabajo está presentado en dos etapas, la primera que consiste en mejorar los datos con un pre-procesamiento de datos aplicando métodos como *garbage in*, *garbage out* utilizados generalmente en proyectos de minería de datos, eliminando instancias del conjunto de datos.

La segunda etapa consiste en el uso del nuevo conjunto de datos para generar un árbol de clasificación usando el algoritmo TreeJ48, mostrando la matriz de confusión para calcular métricas como precisión, exactitud y F_1. Al tener menos información irrelevante se genera un modelo de predicción más exacto, lo cual logró subir la precisión a un 78.17%. Demostrando que el pre-procesamiento de datos mejora la clasificación de instancias del conjunto de datos.

Los autores del trabajo [4] proponen varios algoritmos utilizados en la rama de minería de datos como: SMO (*Sequential Minimal Optimization*), *random forest*, *tree* J48 y *Naive Bayes* para comparar el rendimiento de los algoritmos de clasificación y poder determinar que algoritmo posee una mayor exactitud al realizar el diagnóstico.

Los autores aplican métodos para la limpieza de los datos no requeridos para el estudio, así mismo los autores interpretan los datos faltantes del conjunto de datos. Para la evaluación de los algoritmos utilizan el método conocido como *Cross-validation* junto a las métricas precisión, exactitud y F_1, en donde dividen su conjunto de datos

con una relación 50:50. Los autores mencionan que la relación que utiliza para su conjunto de datos no es ideal, ya que para este tipo de evaluaciones es mejor seccionar en tres partes el conjunto de datos. En los resultados y conclusiones del trabajo nos hace la mención de la exactitud del algoritmo J48, llegando en el mejor de los casos al 73.82% de exactitud al clasificar.

En el 2015 los autores del trabajo [5] presentaron una comparación entre diferentes algoritmos de minería de datos utilizando un conjunto de datos llamado *Pima Indians Diabetes Dataset*. El cual consiste de 768 registros con ocho atributos (edad, insulina, presión, entre otros) y un campo para clasificar (positivo y negativo).

Los algoritmos utilizados en el trabajo son: árboles de decisión J48, *Naive Bayes* y RBF (*Radial Basis Function*), donde utilizan tres métricas de evaluación (precisión, exhaustividad y F_1) para medir el resultado de la clasificación. En el trabajo la precisión más alta fue del algoritmo J48 con un 77.1% en promedio, pero con una clasificación de instancias menor al algoritmo de *Naive Bayes*.

Mencionan los atributos del conjunto, pero no muestran algún tipo de tratamiento de datos. El conjunto de datos fue dividido para realizar el entrenamiento y evaluación, contando con una cantidad aproximada de 230 registros para la evaluación.

En el 2017 Yamilé et al. [2] realizaron un estudio transversal con diseño muestral aleatorio, para detectar la prevalencia de enfermedades crónicas no trasmisibles y sus factores de riesgo. El trabajo utilizó un total de 2085 registros de personas entre 14 municipios, de diferentes edades (32-56 años) utilizando variables como: sexo, edad, perímetro abdominal, glucosa, insulina, triglicéridos, colesterol, entre otros.

Con el uso de medias y desviación estándar para generalizar los atributos presentaron las tablas a varios expertos del campo para diagnosticar cada registro. Llegando a la conclusión que a mayor edad ($\geq$ 50 aproximadamente) se producen cambios hormonales y metabólicos que afectan a varios sistemas. Consecuentemente, desarrollando intolerancia a la glucosa, DMT2 y obesidad abdominal.

En el año 2018 Orlando A. Chan et al. [6] presentaron una investigación realizada sobre un conjunto de datos de 768 pacientes, donde todos los registros son basados en mujeres para la detección de diabetes gestacional, mencionan que dichos atributos en el conjunto de datos son de alta importancia para la detección de DMT2.

Algunas de las variables consideradas son: glucosa, insulina, presión sanguínea y edad. El objetivo final de los autores fue crear un sistema experto para detectar diabetes a partir de los atributos seleccionados del conjunto de datos utilizando algoritmos de clasificación proporcionadas por la herramienta WEKA y BigML.

Los autores utilizan como clasificador árboles de decisión y obtienen un 70% de precisión en la clasificación de pacientes que no presentan DMT2, un 63% para los que si presentan y un 73.83% de exactitud.-En este trabajo realizaremos un tratamiento de datos faltantes a un conjunto de datos con el fin de utilizarlos en mejorar la clasificación usando el algoritmo J48.

Posteriormente se eliminan atributos no relevantes del conjunto utilizando análisis de componentes principales (PCA) y se realiza una comparativa de los resultados, para demostrar que es factible el uso de tratamiento de datos y la reducción de términos con PCA sin perder exactitud en la clasificación.

3. Metodología

El proyecto se divide en cuatro etapas que se resumen a continuación.

3.1. Extracción de datos y análisis de datos

Para esta sección se presenta el conjunto de datos conocido como *Pima Indians Diabetes Dataset*, el cual es utilizado en artículos dentro de la sección de trabajos relacionados. Este conjunto es una recopilación de datos clínicos de mujeres con ascendencia hindú.

Este conjunto de datos será utilizado para la extracción de factores de riesgo, el procedimiento utilizado se compone de tres fases: La primera fase es la recopilación del conjunto de datos, detección de datos anormales y análisis de dichos datos. La segunda fase es explicada en la siguiente sección y en la fase final se hace uso de un análisis de componentes principales con el objetivo de determinar los factores de riesgo con mayor relevancia para llevar a cabo una clasificación.

3.2. Pre-Procesamiento

Con la finalidad de evitar una clasificación pobre que sacrifique la exactitud o precisión de nuestros resultados, el pre-procesamiento de datos es utilizado para eliminar atributos no relevantes, descartando la variedad de estos mismos, ya que los algoritmos empiezan a clasificar erróneamente al tener una gran cantidad de atributos no relevantes dentro de la información recopilada.

Una propuesta muy recurrida de aplicar un pre-procesamiento, es tomar en cuenta los registros completos de datos completos, pero esto provocaría la omisión de muchos datos y no proporcionaría una buena clasificación.

Una mejor propuesta para esta fase es la sustitución por media, la cual funciona calculando el promedio de cada atributo y remplazando a los datos faltantes, dependiendo del atributo que clasifica al registro. Para la aplicación de esta fase se hace uso del lenguaje de programación Python el cual nos ofrece varias herramientas para manipular grandes cantidades de datos de manera eficiente para concluir en un archivo separado por comas (*csv*) donde se guardarán los nuevos datos.

3.3. Algoritmos de clasificación

El objetivo de la implementación de los algoritmos de clasificación es descubrir que atributos son los encargados de realizar la clasificación de los datos para comparar estos atributos con los resultados de PCA y determinar una lista de atributos principales proveniente de los factores de riesgo.

Algunos algoritmos de clasificación utilizados más frecuentemente son: SVM (*Support Vector Machine*) [7], *Random Forest* [8], Arboles de decisión J48 [9] y *Naive Bayes* [10]. Una vez obtenido los factores de riesgos más relevantes del conjunto de datos se realizará nuevamente la clasificación del conjunto de datos para ver si existe una mejora en su clasificación con menor cantidad de atributos.

Tabla 1. Descripción del conjunto de datos.

Atributo	*Descripción*	*Min-Max*
Embarazos	Cantidad de embarazos	0-17
Glucosa	Concentración de glucosa en plasma a dos horas en una prueba de tolerancia a la glucosa oral	0-199
Presión sanguínea	Presión arterial diastólica (mm Hg)	0-122
Grosor de la piel	Espesor del pliegue cutáneo del tríceps (mm)	0-99
Insulina	Insulina sérica de 2 horas (mu U / ml)	0-846
IMC	Índice de Masa Corporal (Kg/m^2)	0-67.1
PedigreeFunction	Función de árbol genealógico de la diabetes	0.08-2.42
Edad	Edad en años	21-81
Resultado	Resultado del paciente a la enfermedad de diabetes	0-1

3.4. Evaluación

La evaluación se puede realizar de dos formas diferentes. La primera consiste en el uso de métricas de evaluación para medir los resultados obtenidos de la clasificación. La segunda consiste en utilizar un experto en el campo, el cual se encargará de revisar los estudios del paciente y dar el veredicto si la clasificación es correcta.

Para la evaluación es necesario tener presente la matriz de confusión generada por los algoritmos de clasificación, la cual nos compara la predicción de las clases con los resultados etiquetados, resultando en cuatro métricas [11]:

- Precisión: Es el número de casos relevantes recuperados entre el número de casos recuperados (ver Ecuación 1).
- Exhaustividad: El cual nos informa sobre la capacidad para identificar nuevos registros usando el modelo matemático generado (ver Ecuación 2).
- Exactitud: Mide el porcentaje de casos que el modelo ha acertado o clasificado correctamente (ver Ecuación 3).
- F_1: Es utilizado para combinar las medidas de precisión y exhaustividad en un solo valor (ver Ecuación 4):

$$Precisión = \frac{VP}{VP+FP}, \tag{1}$$

$$Exhaustividad = \frac{VP}{\mathrm{VP}+FN}, \tag{2}$$

$$Exactitud = \frac{VP+VN}{VP+VN+FP+FN}, \tag{3}$$

$$F_1 = 2 \cdot \frac{Precisión \cdot \mathrm{Exhaustividad}}{Precisión + \mathrm{Exhaustividad}}. \tag{4}$$

donde:

Verdaderos positivos (VP): Son resultados positivos clasificados correctamente.

Verdaderos negativos (VN): Son resultados negativos clasificados como positivos.

Falsos positivos (FP): Son resultados negativos clasificados como positivos.

Falsos negativos (FN): Son resultados negativos clasificados correctamente.

Tabla 1. Resultados experimentales.

Resultados	*Datos Completos*	*Tratamiento de datos*	*Tratamiento de datos + PCA*
Correctamente clasificado	567	666	**668**
Incorrectamente clasificado	201	102	**100**
Error absoluto (promedio)	31.58%	16.58%	**16.41%**
Precisión	63.24%	82.42%	**83.87%**
Exhaustividad	59.70%	**78.73%**	77.61%
Exactitud	73.83%	86.72%	**86.98%**
F_1	61.42%	80.53%	**80.62%**

El objetivo de la evaluación es de medir la eficiencia del modelo creado en registros nuevos. Esta eficiencia es medida en porcentajes que pueden variar dependiendo del conjunto de datos utilizado.

4. Resultados

En esta sección se presenta el conjunto de datos utilizados en los experimentos y los resultados obtenidos.

4.1. Conjunto de datos

El conjunto de datos utilizado en el estado del arte está basado en mediciones medicas provenientes del Instituto Nacional de Diabetes y Enfermedades Digestivas y Renales, donde el objetivo es predecir si un paciente tiene diabetes o no basándose en determinadas medidas de un diagnóstico. En particular, todos los pacientes en este conjunto de datos son mujeres de al menos 21 años y de ascendencia hindú, con un total de 500 registros no diabéticos y 268 diabéticos; también conocida como *Pima Indians Diabetes Dataset* [12].

En la Tabla 1 se describe el conjunto de datos que incluye ocho atributos y un resultado que indica si el paciente padece diabetes (clase 1) o no (clase 0).

4.2. Resultados experimentales

Antes de tratar los datos se realiza una ejecución del algoritmo de clasificación Treej48, para obtener las métricas de clasificación antes de cualquier modificación al conjunto de datos. Una vez realizado la clasificación se aplica el pre-procesamiento de los datos y se ejecuta nuevamente una ejecución del clasificador, dando dos conjuntos de datos extras para la comparación de los resultados al reducir los atributos.

Para aplicar la compresión de datos (PCA) se hace uso de la herramienta *Pandas* del lenguaje *Python*, el cual nos proporciona un *DataFrame* que representa una estructura ordenada de los datos y podemos crear arreglos de una manera más sencilla a partir de

I	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	169.5	33.6	0.627	50	1
1	1	85	66	29	102.5	26.6	0.351	31	0
2	8	183	64	32	169.5	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Fig. 1. Orden del archivo *CSV*.

```
Nombre del archivo (.csv): Diabetes
Columna de inicio: 1
Columna final: 9
Encabezados (Separados por comas): Pregnancies,Glucose,BP,ST,Insulin,BMI,DPF,Age
```

Fig. 2. Ejecución de algoritmo de reducción de términos.

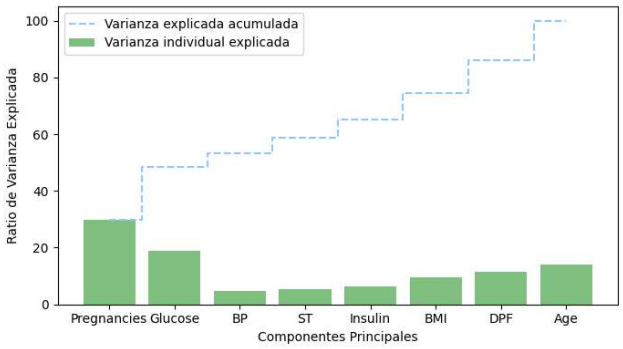


Fig. 2. Resultado final del algoritmo PCA.

un *csv* [13]. El archivo es ordenado respetando una única regla, la etiqueta clasificadora (*Outcome*) debe ser la última columna del archivo como se muestra en la Fig. 1.

Una vez que el archivo fichero cumpla con esa única regla será la entrada en un *script* de *Python*, donde se realizará la ejecución del algoritmo de análisis de componentes principales. También se solicitan datos como: Columna de inicio, columna final y los encabezados para la ejecución de los algoritmos. Posteriormente el *script* muestra una salida con la ponderación de cada atributo (ver Fig. 2).

En la Fig. 3 se puede ver que el algoritmo PCA utiliza la varianza explicada para representar la importancia de cada atributo dentro del conjunto utilizado y la varianza acumulada, la cual es la suma de cada columna, la cual al final siempre tiene que ser 100%. Los datos con mayor porcentaje son los atributos de embarazos y la glucosa, representado un 50% de la varianza dentro del conjunto de datos, aproximadamente. Los siguientes atributos a tomar en cuanta: edad, la función de árbol genealógico de la diabetes, índice de masa corporal e Insulina a con la finalidad de lograr un 90% de representación de los datos. Dejando a la presión arterial y grosor de la piel un 10% de representación.

Para llevar a cabo la clasificación del conjunto de datos se hace uso de la herramienta *WEKA*, ya que esta nos da una gran gama de algoritmos utilizados en el entorno de minería de datos, los cuales utilizan algoritmos de clasificación [14].

En las columnas de la Tabla 2, se presentan los resultados obtenidos por cada conjunto de datos usando validación cruzada de 10 *folds*, en donde la primera columna

Tabla 2. Comparativa de clasificación y métricas.

Tratamiento	Fuente	Precisión	Exhaus-tividad	Exactitud	F_1
Ninguno	[6]	63.24%	59.70%	73.83%	61.42%
Desconocido	[5]	78.60%	77.20%	N/R	77.10%
Pre-procesa-miento	Propia	82.42%	**78.73%**	86.72%	80.53%
Pre-procesa-miento + PCA	Propia	**83.87%**	77.61%	**86.98%**	**80.62%**

(datos completos) es el conjunto de datos sin retirar ningún atributo; la segunda columna (tratamiento de datos) representa el conjunto de datos tratando usando sustitución por media y la última columna representa el análisis de componentes principales sobre el conjunto de datos con tratamiento de datos y retirando atributos con el ~~los~~ análisis de PCA.

Como se puede ver en la Tabla 2, el aumento en los datos ***correctamente clasificados con tratamiento de datos*** (99 instancias) es un indicativo de una mejora en la clasificación del conjunto de datos utilizado que se aprecia en el incremento en la ***precisión***, ***exactitud*** y F_1.

Al comparar los resultados de las columnas tres y cuatro se observa que el incremento es mínimo (solo 2 instancias), pero demuestra que los atributos eliminados (grosor de piel y presión sanguínea), quienes dieron un bajo porcentaje reportado en el análisis de componentes principales realizado con anterioridad (>10%) no representan importancia al momento de utilizar el algoritmo de clasificación.

La Tabla 3 nos da una vista de los resultados en donde la primera columna se presentan la fuente, y las columnas restantes corresponde a cada métrica explicada con anterioridad. Se puede observar que la diferencia en las métricas reportadas por los autores del trabajo [6] y las presentadas en este trabajo existe un aumento en cada métrica con un tratamiento de datos basado en promedios, llegando a tener una precisión del 82.42% y una exactitud del 86.72%.

Mientras que la reducción de dos atributos en el conjunto de datos ayudó significativamente a mejorar la precisión llegando a un 83.87% y la exactitud al 86.98%. Mostrando que los atributos eliminados no tenían relevancia para la clasificación.

5. Conclusiones

El uso del algoritmo PCA, que es de suma importancia en el campo de la minería de datos, permitió reducir la cantidad de atributos utilizados en la clasificación. Esto provocó una ejecución más rápida dado que hay una cantidad menor de información en el conjunto de datos. Los resultados experimentales muestran una mejora en los resultados de las métricas de exactitud (86.92%), precisión (83.87%) y F1 (80.62%); en un total de 668 registros.

Al utilizar los algoritmos de clasificación con un pre-procesamiento de datos podemos ser capaces de ver que atributos son relevantes al momento de realizar una clasificación, pero al comprimir los datos después de tratar la información nos permite

eliminar atributos sin perder confianza en el modelo generado, logrando el 86.98% de exactitud.

El usar un tratamiento de datos ayuda a mejorar la clasificación del algoritmo por si solo en la mayoría de los casos, en este trabajo se utilizó un método simple para tratar datos no válidos, dejando la posibilidad de utilizar diferentes métodos como *k* vecinos más próximos o realizar discretización de datos para mejorar aún más la clasificación.

Las mejoras no simplemente se encuentran en el tratamiento de datos, sino que también aplican a los algoritmos de clasificación, en donde métodos más avanzados como son las redes neuronales o clasificadores ingenuos como *Naive Bayes* pueden aumentar la precisión de registros correctamente clasificados, ya que algunos se basan en reglas establecidas por el conjunto de datos sacando conclusiones de los resultados.

Referencias

1. Rodríguez-Rivera, N. S., Cuautle-Rodríguez, P., Castillo-Nájera, F., Molina-Guarneros, J. A.: Identification of genetic variants in pharmacogenetic genes associated with type 2 diabetes in a mexican-mestizo population. Biomed. Reports, vol. 7, no. 1, pp. 21–28 (2017) doi: 10.3892/br.2017.921
2. Miguel, P. E., Sarmiento, Y., Mariño, A. L., Llorente, Y., Rodríguez, T., Peña, M.: Prevalencia de enfermedades crónicas no transmisibles y factores de riesgo en adultos mayores de Holguín. Rev. Finlay, vol. 7, no. 3, pp. 155–167 (2017)
3. Al-Jarullah, A. A.: Decision tree discovery for the diagnosis of type II diabetes. In: Proceeding of International Conference on Innovations in Information Technology, pp. 303–307 (2011) doi: 10.1109/INNOVATIONS.2011.5893838
4. Hemant, P., Pushpavathi, T.: A novel approach to predict diabetes by cascading clustering and classification. In: Proceedings of Third International Conference on Computing Communication and Networking Technologies (2012) doi: 10.1109/ICCCNT.2012.6396069
5. Sa'di, S., Maleki, A., Hashemi, R., Panbechi, Z., Chalabi, K.: Comparison of data mining algorithms in the diagnosis of type II diabetes. International Journal on Computational Science & Applications (IJCSA), vol. 5, no. 5, pp. 1–12 (2015) doi: 10.5121/ijcsa.2015.5501
6. Chan, O., Peña, J., Vianne, J., Zapata, M.: Construcción de un modelo de predicción para apoyo al diagnóstico de diabetes (construction of a prediction model to support the diabetes diagnosis). vol. 40, no. 130, pp. 2105–2122 (2018)
7. Joaquín-Amat, R.: Máquinas de vector soporte (SVM) con python. Cienciadedatos.net (2020) https://www.cienciadedatos.net/documentos/py24-svm-python.html
8. Joaquín-Amat, R.: Random forest con python. Cienciadedatos.net (2020) https://www.cienciadedatos.net/documentos/py08_random_forest_python.html
9. Joaquín-Amat, R.: Árboles de decisión con python: Regresión y clasificación (2020) https://www.cienciadedatos.net/documentos/py07_arboles_decision_pyt hon.html
10. Pedregosa, F.: Scikit-learn: Machine learning in python. The Journal of machine Learning research, vol. 12, pp. 2825–2830 (2011)
11. Heras, J. M.: Precision, recall, F1, accuracy en clasificación. IArtificial.net (2020) https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/
12. Machine Learning: Pima indians diabetes database. Predict the onset of diabetes based on diagnostic measures (2016) https://www.kaggle.com/uciml/pimaindians-diabetes-database
13. Wade, R.: Reading CSV files. Advanced analytics in power BI with R and python: ingesting, transforming, visualizing, apress (2020) pp. 151–175

14. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H.: The WEKA data mining software: An update. ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10–18 (2009) https://www.kdd.org/exploration_files/ p2V11n1.pdf

ISSN 1870-4069

Detección de lugares disponibles en un estacionamiento

Víctor Romero-Bautista[1], Aldrin Barreto-Flores[1], Salvador E. Ayala-Raggi[1], Verónica E. Bautista-López[2]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Electrónica, México

[2] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, México

victor.romerobau@alumno.buap.mx, {aldrin.barreto, salvador.raggi, veronica.bautistalo}@correo.buap.mx

Resumen: Los sistemas de asistencia a estacionamientos basados en visión por computadora, cada vez son empleados con mayor frecuencia, para el mejoramiento del tránsito vehicular en el área urbana. Estos sistemas proporcionan información de la disponibilidad de un estacionamiento, además, contribuyen a tener una mejor organización en el mismo y a reducir el tiempo de búsqueda de un espacio disponible a los usuarios. El principal reto de estos sistemas se debe a la variación de iluminación que suele ocurrir durante el día, ya que puede presentarse una disminución de iluminación debido a efectos de sombra, o bien, incrementos a causa del sol, además de éstos, se pueden presentar otros inconvenientes, como variación de tonalidad en los autos, o la oclusión entre los mismos. En este trabajo se propone un método para la detección de lugares disponibles en un estacionamiento, en el cual se emplea un procedimiento para la ubicación semiautomática de regiones de interés (ROI). Se utilizó el modelo de intersección cortical (ICM) para la extracción de características, el análisis de componentes principales (PCA) para reducción de características y el perceptrón multicapa para la etapa de clasificación. El método propuesto presentó el 90% de efectividad en la detección, presentando baja susceptibilidad a los cambios de iluminación.

Palabras clave*:* Aprendizaje, detección, lugares de estacionamiento, firma, ROI.

Detection of Available Places in a Parking Lot

Abstract. Parking assistance systems based on computer vision are used more and more frequently to improve vehicular traffic in urban areas. These systems provide information on the availability of a parking lot, in addition, they contribute to having a better organization in it and to reducing the search time for an available space for users. The main challenge of these systems is due to the variation in lighting that usually occurs during the day, since there may be a decrease in lighting due to shadow effects, or increases due to the sun, in addition

to these, there may be other inconveniences, such as variation of tonality in the cars, or the occlusion between them. In this work, a method for the detection of available spaces in a parking lot is proposed, in which a procedure for the semi-automatic location of regions of interest (ROI) is used. Cortical Intersection Model (ICM) was used for feature extraction, Principal Component Analysis (PCA) for feature reduction, and multilayer perceptron for the classification stage. The proposed method presented 90% effectiveness in detection, presenting low susceptibility to lighting changes.

Keywords: Learning, detection, parking spots, signature, ROI.

1. Introducción

La detección de lugares disponibles en estacionamientos basados en visión por computadora, presentan diversas dificultades, para lograr una adecuada detección. Algunas de éstas, corresponden a los cambios de iluminación que ocurren a lo largo del día, efectos de sombra, diferentes tonalidades de color que pueden presentar los automóviles, oclusión entre los objetos de interés, además de otros. La detección de lugares disponibles en estacionamientos es comúnmente empleada en los sistemas de asistencia a estacionamientos, los cuales proporcionan información a los usuarios acerca de la disponibilidad de un estacionamiento, además, contribuyen a mejorar el flujo vehicular y la organización dentro del mismo.

Algunos trabajos propuestos en el estado del arte para la detección de lugares disponibles en estacionamientos, basan su operación en la detección de automóviles, como en [1], donde se propone un método para obtener las características de imágenes de autos, mediante el histograma de gradientes orientados (HOG), y para llevar a cabo la detección de éstos, emplea máquinas de vectores de soporte (SVM); el mismo enfoque se da en [2], donde se propone un método basado en la resta del fondo, y posteriormente la segmentación de la imagen resultante, usando la red neuronal pulso-acoplada (PCNN), para así llevar a cabo la detección de los autos.

Por otro lado, se encuentran los métodos que emplean la detección de bordes y esquinas, los cuales corresponden a las marcas que delimitan los espacios de estacionamiento, como es el caso de [3], el cual emplea el detector de esquinas de Harris.

También se encuentran los métodos que basan su operación en la ubicación de regiones de interés (ROI), éstos normalmente indican de forma manual las coordenadas de cada ROI en la imagen, como en [4], donde una vez generadas las ROI, aplica la resta del fondo a cada una de éstas, y para el reconocimiento de los objetos, se extrae el HOG, detector de esquinas mediante la transformada de características invariantes (SIFT) y métricas de los espacios de color YUV, HSV y YCrCb; en [5], de igual manera se aplica la ubicación de ROI señalando sus coordenadas manualmente, y para la detección de lugares disponibles, aplica un umbralado a cada ROI.

Métodos recientes para la detección de lugares disponibles en estacionamientos, llevan a cabo el uso de redes neuronales convolucionales (CNN), como en [6], donde se emplea la CNN para generar el *bounding boxing* en los automóviles; en [7] se emplea la CNN, para la extracción de características, y para clasificación se hace uso de SVM.

Estos trabajos presentan algunas ventajas para ciertos casos, pero también, son susceptibles a inconvenientes. Por ejemplo, en [1], [2], [4] y [7], se presenta un bajo rendimiento ante la presencia de sombras y variación de iluminación. Por otro lado, en [5], se tiene una respuesta mejor a cambios de iluminación, sin embargo, éste únicamente fue probado a escala. En [6], se presentó una baja efectividad, debido a que se utilizó un conjunto de datos limitado, y el trabajo presentado en [3], demanda que las líneas delimitadoras de los espacios de estacionamiento se encuentren uniformemente remarcadas.

En este trabajo se propone un método para la detección de lugares disponibles en un estacionamiento, en el cual se emplea un procedimiento para la ubicación semiautomática de ROI, por otro lado, para el reconocimiento de los objetos, se hace uso de la firma, que genera la red ICM, para la cual, se propone el valor de sus parámetros. Además, se lleva a cabo la reducción de características, mediante el PCA y para la clasificación de éstas, se hace uso de la red MLP.

El método propuesto se evaluó con un conjunto de 210 imágenes con cuatro tomas diferentes de topologías de estacionamiento, y se implementó en una Raspberry Pi 4. Se obtuvo una efectividad del 90% y un tiempo promedio de ejecución de 2 segundos.

2. Método propuesto

La propuesta del método consiste de los siguientes cuatro pasos: generación de regiones de interés (ROI), extracción de características, reducción de dimensionalidad y detección. A continuación, se presenta más a detalle cada uno de estos pasos.

2.1. Generación de regiones de interés (ROI)

En este primer paso, se generan las zonas donde se encuentran los lugares de estacionamiento. En algunos trabajos, como en [4] y [5], se lleva a cabo esta etapa, indicando las coordenadas de cada una dentro de la imagen, demandando tiempo importante para esta tarea.

Con el objetivo de disminuir el tiempo de ubicación de ROI, se optó por llevar a cabo la localización de éstas de manera semiautomática, de tal forma que, únicamente se requiera indicar la cantidad lugares de estacionamiento para ciertas zonas en la imagen.

De acuerdo con la distribución de los lugares de estacionamiento en el conjunto de imágenes utilizado para este trabajo, se identificaron dos zonas (superior e inferior), en las cuales se ubican los espacios de estacionamiento, éstas se presentan en la figura 1.

Partiendo de la distribución presentada en la figura 1, se diseñó un método para generar las ROI de forma semiautomática, en el cual únicamente se debe indicar la cantidad de lugares de estacionamiento que se encuentran en cada zona, considerando que la cámara de captura no cambie de posición.

Este método consiste, primeramente, en dividir la imagen en tres filas, como se presenta en el inciso (a) de la figura 2, las cuales se denominan: fila superior, intermedia e inferior. La altura de las filas superior e inferior corresponde al tamaño en alto del espacio de estacionamiento, mientras que la fila intermedia es el área restante y no es considerada.

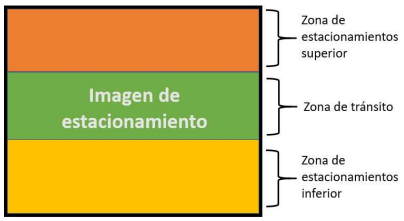
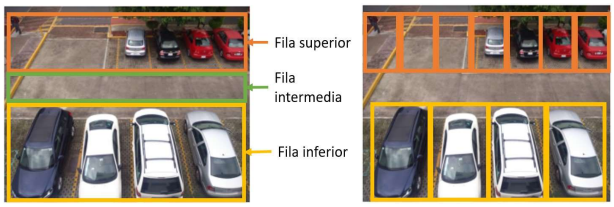


Fig .1. Representación de distribución de lugares de estacionamiento en el conjunto de imágenes.



(a) (b)

Fig. 2. Localización de regiones de interés; (a) generación de filas; (b) generación de recuadros *(bounding boxing)*.

(a) (b)

Fig. 3. Resultados de aplicar el método propuesto para la localización de ROI.

En seguida, las filas superior e inferior son divididas en columnas, de acuerdo al número de espacios de estacionamiento que se considere pueden ser contenidos dentro de cada fila, de tal manera que, cada columna pueda generar un recuadro que contenga un espacio de estacionamiento.

Como resultado, se presenta la imagen del inciso (b) de la figura 2, donde la fila inferior se ha dividido en cuatro columnas y la superior en siete, ya que, con esto, se logra contener un lugar de estacionamiento en cada recuadro generado.

El número de columnas a dividir es el parámetro que permite generar las ROI, y éste es establecido de acuerdo al tipo de toma con la que se esté trabajando.

Una vez hecho lo anterior, dependiendo de la topología del estacionamiento, se puede presentar una región que no se debe considerar, como se muestra en el inciso (a) de la figura 3, donde el recuadro en color rojo corresponde a la entrada del estacionamiento, por lo tanto, los recuadros generados dentro de esta región no deben ser tomados en cuenta.

Para esto, se estableció un parámetro, el cual indica si se deben ignorar algunos recuadros generados en cada fila de estacionamientos, que para el caso de la imagen del inciso (a) de la figura 3, se establece que los primeros tres recuadros (de izquierda a derecha), de la fila superior se deben saltar, obteniendo como resultado la imagen del inciso (b) de la figura 3, la cual muestra las regiones de interés generadas.

2.2. Extracción de características

En este paso, por cada ROI generada, se lleva a cabo la extracción de la firma ($G[n]$) producida por la red ICM, y a su vez, es almacenada en el vector de características a. La red ICM es un modelo simplificado de la red neuronal pulso-acoplada (PCNN) [8, 9], este tipo de red neuronal es ampliamente utilizada en el procesamiento digital de imágenes, como en: segmentación [10], reducción de ruido [11], detección de bordes [12], extracción de características [13], entre otras aplicaciones.

En las ecuaciones (1), (2) y (3), se presenta el modelo de red ICM, las cuales corresponden respectivamente a, el potencial interno (F), umbral dinámico (θ) y salida (Y), para una neurona i, j, donde S es la imagen de entrada y W la matriz de pesos sin ‘apticos con tamaño k, l. Para el correcto funcionamiento de la ICM, se requiere del ajuste de los parámetros f, g y h.

La extracción de características con la red ICM, en general para una PCNN, se lleva a cabo mediante la obtención de la firma ($G[n]$), ésta cuantifica la cantidad de neuronas activadas por cada pulso generado durante su operación, y se expresa por la ecuación (4). Se llevaron a cabo una serie de pruebas con diferentes valores de parámetros, donde se concluyó que es posible obtener un patrón de firma adecuado, con una cantidad de 50 pulsos, empleando el valor de los parámetros: $f = 0{,}1$; $g = 0{,}8$; $h = 20$; y W como se presenta en la ecuación (5). La firma de las ROI, es generada tomando en cuenta los canales r, g y b, de la imagen:

$$F_{ij}[n] = fF_{ij}[n-1] + \sum_{kl} W_{ijkl} Y_{kl}[n-1] + S_{ij} \tag{1}$$

$$\theta_{ij}[n] = g\theta_{ij}[n-1] + hY_{ij}[n-1], \tag{2}$$

$$Y[n] = \begin{cases} 1 \; si \; fF_{ij}[n] > \; \theta_{ij}[n], \\ 0 \; en \; caso \; contrario, \end{cases} \tag{3}$$

$$G[n] = \; \sum_{i,j} Y_{i,j}[n], \tag{4}$$

$$W = \begin{pmatrix} 0{,}25 & 0{,}5 & 0{,}25 \\ 0{,}5 & 0 & 0{,}5 \\ 0{,}25 & 0{,}5 & 0{,}25 \end{pmatrix}. \tag{5}$$

2.3. Reducción de dimensionalidad

Una vez que se ha generado el vector a, éste contendrá un total de 50 componentes, los cuales son normalizados a valores entre cero y uno, mediante la ecuación (6), generando así, el vector normalizado a_{norm}. Posteriormente se aplica la reducción de

Fig. 4. Tomas correspondientes a las cuatro topologías de estacionamiento empleadas para la evaluación del método propuesto.

Fig. 5. Resultado de ubicación de regiones de interés.

dimensionalidad a dos componentes ($C1$ y $C2$), mediante el PCA [14], obteniendo el vector x, representado en la ecuación (7):

$$a_{norm} = \frac{a}{max(a)}, \tag{6}$$

$$x = [C1, C2]. \tag{7}$$

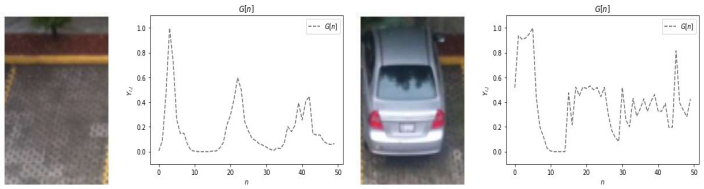
2.4. Detección

Finalmente, para realizar la detección de los lugares disponibles, se lleva a cabo la clasificación del vector x, el cual es introducido a una red perceptrón multicapa (MLP) previamente entrenada [15], donde la salida de la red es enviada a una función de activación softmax [16], la cual es representada por la ecuación (8):

Algoritmo 1: Algoritmo propuesto para la detección espacios disponibles en un estacionamiento.

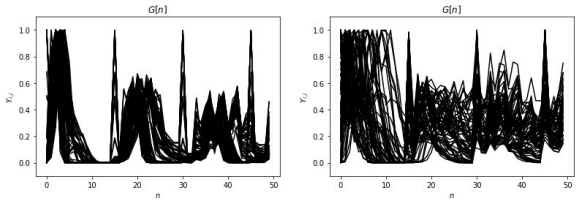
Input: I_{rgb}
Result: Imagen con ubicación de los espacios disponibles y ocupados I_r

```
Ubicar las regiones de interés (ROI) en I_rgb;
for cada ROI do
    Generar firma  G[n] → a;
    Generar a_norm  Y reducer a dos componentes  → x;
    Clasificar x  Mediante la red MLP  → σ;
    if σ[0, 1] > 80 then
        Marcar ROI como ocupado;
    end
    if σ[1, 0] > 80 then
        Marcar ROI como disponible;
    end
end
```



(a) (b) (c) (d)

Fig. 6. Resultados de extracción de la firma G[n].



(a) (b)

Fig. 7. Conjunto de firmas obtenidas de las imágenes de aprendizaje; (a) firmas generadas por espacios disponibles; (b) firmas generadas por espacios ocupados.

$$\sigma(y_k) = \frac{e^{y_k}}{\sum_{l=1}^{c} e^{(y_l)}}, k = 1, ..., c. \tag{8}$$

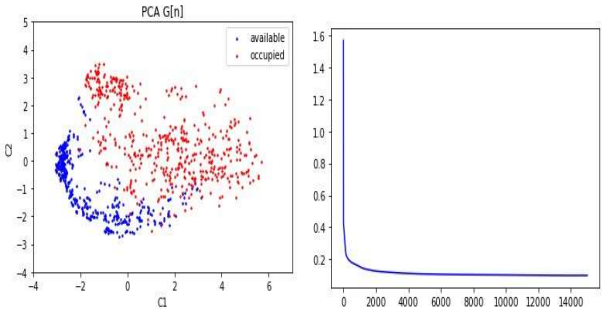
donde σ corresponde a la salida de la red, dado en porcentaje de reconocimiento; y es la salida de una neurona en la capa de salida; k corresponde a la neurona actual y c a la cantidad de neuronas en la capa de salida.

Se estableció la etiqueta $[1,0]$ para identificar a la ROI que corresponda a espacios disponibles, y $[0,1]$ para ocupados. Donde la ROI será marcada como ocupada o disponible, si la salida de la red es mayor al 80%. En el algoritmo 2.1, se presentan el método propuesto para este trabajo.

3. Resultados

La intención de este trabajo es implementarlo en un estacionamiento, sin embargo, debido a la situación que se ha vivido por la pandemia, ésto no fue posible. Por lo anterior, se llevaron a cabo pruebas experimentales con 300 imágenes, las cuales contienen cuatro tomas de estacionamientos con topologías distintas (ver figura 4). Se utilizaron para la etapa de entrenamiento, un total de 90 imágenes, y las 210 restantes para evaluación.

A continuación, se presentan los resultados de la ubicación de ROI, extracción de características, reducción de componentes, etapa de aprendizaje y evaluación del método propuesto.



(a) (b)

Fig. 8. Resultados de reducción de componentes y aprendizaje del clasificador MLP.

Tabla 1. Resultados obtenidos por el método propuesto y en trabajos del estado del arte.

Autores	Wahyono [1]	Tatulea [4]	Bibi [5]	Suhr [3]	Banerjee [6]	Acharya [7]	Propuesto
Muestras de evaluación	2,656	106,192	200	134	2,000	504,139	2,009
Efectividad (%)	98.7	94.41	98	95	71.2	98.2	90
T. ejecución (segs)	-	-	-	-	-	2	2
Vel. de CPU (GHz)	-	-	-	-	-	2.5	1.5

a) b) c) d)

Fig. 9. Resultados de la implementación del algoritmo propuesto en la tarjeta Raspberry pi 4.

3.1. Obtención de regiones de interés

En la figura 5, se muestra el resultado obtenido del método propuesto de ubicación de regiones de interés, para cada topología de estacionamiento, donde son resaltadas en color rojo las ROI generadas.

3.2. Extracción de firma ($G[n]$)

En la figura 6, se presentan los resultados de la extracción de la firma, donde la imagen del inciso (a), corresponde a un espacio disponible y la gráfica del inciso (b)

presenta su firma, por otro lado, la imagen del inciso (c) presenta un espacio ocupado por un automóvil, y la gráfica del inciso (d) indica su firma respectiva.

De acuerdo con las 90 imágenes utilizadas para la etapa de aprendizaje, se generaron un total de 940 muestras, de las cuales 498 pertenecen a espacios ocupados y 456 son de espacios disponibles. En las gráficas de la figura 7, se presenta el total de firmas recabadas de las imágenes de entrenamiento, donde el inciso (a) muestra el total de firmas obtenidas por imágenes de lugares disponibles, mientras que el inciso (b) presenta el total de firmas obtenidas por lugares ocupados.

3.3. Reducción de componentes y aprendizaje

Partiendo del conjunto de firmas generado por las imágenes de aprendizaje, se llevó a cabo la reducción de dimensionalidad de éstos, a dos componentes, mediante el método PCA, el resultado se presenta en la gráfica del inciso (a) de la figura 8.

Posteriormente, este conjunto de muestras se utilizó para el entrenamiento del clasificador MLP, el cual se configuró a una arquitectura piramidal, estableciendo dos capas ocultas, con once neuronas en la primera capa, nueve en la segunda y dos neuronas en la capa de salida, conectadas a una función de activación softmax. El entrenamiento del clasificador se llevó a cabo mediante el algoritmo *backpropagation*.

En la gráfica del inciso (b) de la figura 8, se presenta el decremento del error obtenido durante la etapa de aprendizaje del MLP.

3.4. Resultados de detección

Para la etapa de evaluación se utilizó un conjunto de 210 imágenes con cuatro topologías de estacionamiento diferentes. El algoritmo 2.1, fue implementado en lenguaje Python, el cual se ejecutó mediante una interfaz de usuario en la tarjeta Raspberry Pi 4, que cuenta con un procesador Quad Core Cortex-A72 con velocidad de 1.5 GHz y 4 GB de memoria RAM.

Se evaluó la velocidad de ejecución, así como, la efectividad, representada en la ecuación (9), donde FN corresponde a los falsos negativos, VP a los verdaderos positivos y M es el número de muestras:

$$efectividad = \frac{FN + VP}{M}. \tag{9}$$

En la tabla 1 se presentan los resultados obtenidos de la evaluación del método propuesto, en conjunto con los alcanzados por otros autores, en trabajos similares. Cabe mencionar que, en las métricas de tiempo de ejecución y velocidad de CPU, algunos autores no incluyeron esta información, por tal motivo, se ha dejado un espacio vacío en la tabla.

Además, en la figura 9, se muestran los resultados de la implementación del algoritmo en la tarjeta Raspberry Pi 4, donde son enmarcados en color verde los lugares disponibles y en rojo los ocupados.

Los resultados obtenidos, presentan una efectividad del 90%, el cual es 18% más, al alcanzado en [6], sin embargo, los trabajos [1, 5, 7], presentan aproximadamente una efectividad 8% mayor al obtenido por el método propuesto, mientras que, los trabajos [4, 3], presentan una efectividad del 5% más, al alcanzado.

Por otro lado, se obtuvo un promedio en velocidad de ejecución por imagen de 2 segundos, este resultado, es similar al generado en [7], el cual obtuvo un 8.2% más de efectividad que el algoritmo propuesto, sin embargo, éste utilizó 1 GHz más en velocidad de CPU.

4. Conclusiones y trabajo futuro

En este trabajo se ha presentado un método para la detección de lugares disponibles en un estacionamiento, el cual plantea la ubicación de ROI de manera semiautomática, considerando cuatro topologías de estacionamiento distintas, por otro lado, se proponen los valores de los parámetros de la red ICM, para emplearla en la extracción de características.

Bajo el método propuesto los resultados obtenidos presentan una efectividad del 90%, lo cual demuestra que es ligeramente robusto, presentando una baja sensibilidad a los cambios de iluminación, así como, a los efectos de sombra. Sin embargo, se presenta una diferencia de efectividad entre 5% y 8% menos, en otros trabajos del estado del arte.

Por otro lado, considerando la cantidad de imágenes empleadas, el trabajo propuesto, presentó resultados favorables, a pesar de contar un conjunto limitado de imágenes, en comparación con otros trabajos como lo son: [1, 4, 5, 6, 7].

Como trabajo futuro, se propone generar de manera automática las ROI a través del análisis de imágenes. Para mejorar la efectividad del algoritmo se propone, aumentar el conjunto de imágenes de estacionamientos, agregar una etapa de preprocesamiento en las ROI, para resaltar sus características.

Referencias

1. Wahyono, W., Hoang, V. D., Jo, K. H.: Multiscale car detection using oriented gradient feature and boosting machine. In: Nguyen, N., T., Trawinski, B., Fujita, H., Hong, T., P. (eds.) Intelligent Information and Database Systems, Springer, vol. 9621, pp. 731–740 (2016) doi: 10.1007/978-3-662-49381-6_70
2. Xu, Y., Wang, S., Li, X.: Vehicle video detection based on pulsed coupled neural network. In: Proceedings of 7th International Congress on Image and Signal Processing, pp. 731–735 (2014)
3. Suhr, J. K., Jung, H. G.: Fully-automatic recognition of various parking slot markings in around view monitor (avm) image sequences. In: Proceedings of 15th International IEEE Conference on Intelligent Transportation Systems, pp. 1294–1299 (2012)
4. Tâtulea, P., Câlin, F., Brad, R., Brâncovean, L., Greavu, M.: An image feature based method for parking lot occupancy. In: Proceedings of Future Internet, vol. 11, no. 8, pp. 169 (2019)
5. Bibi, N., Majid, M. N., Dawood, H., Guo, P.: Automatic parking space detection system. In: Proceedings of Second international conference on multimedia and image processing (ICMIP), pp. 11–15 (2017)
6. Banerjee, S., Ashwin, T., Guddeti, R. M. R.: Automated parking system in smart campus using computer vision technique. In: Proceedings of IEEE Region 10 Conference TENCON´19, pp. 931–935 (2019)

7. Acharya, D., Yan, W., Khoshelham, K.: Real-time image-based parking occupancy detection using deep learning. In: Proceedings of the 5th Annual Conference of Research@Locate, vol. 2087, pp. 33–40 (2018)
8. Ma, Y., Zhan, K., Wang, Z.: Applications of pulse-coupled neural networks (2011)
9. Lindblad, T., Kinser, J. M., Taylor, J. G.: Image processing using pulse-coupled neural networks. Springer, pp. 2 (2013)
10. Li, H., Guo, L., Yu, P., Chen, J., Tang, Y.: Image segmentation based on iterative self-organizing data clustering threshold of PCNN. In: 2nd International Conference on Cloud Computing and Internet of Things (CCIOT), pp. 73–77 (2016)
11. Rangel, E., Lavalle, M., Sossa, H.: Filtrado de ruido gaussiano mediante redes neuronales pulso-acopladas. Computación y Sistemas, vol. 21, pp. 381–395 (2017)
12. Shi, Z., Hu, J.: Image edge detection method based on a simplified PCNN model with anisotropic linking mechanism. In: Proceedings of 10th International Conference on Intelligent Systems Design and Applications, pp. 330–335 (2010)
13. Gu. X.: Feature extraction using unit-linking pulse coupled neural network and its applications. Proceedings of Neural Processing Letters, Springer, vol. 27, no. 1, pp. 25–41 (2008) doi: 10.1007/s11063-007-9057-6
14. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 586–591 (1991)
15. Da-Silva, I. N., Spatti, D. H., Flauzino, R. A., Liboni, L. H. B.: Artificial neural networks. Springer Cham, vol. 39 (2017)
16. Marsland, S.: Machine learning. Second edition. Chapman and Hall/CRC (2015)

ISSN 1870-4069

Linear Control of a Two-Wheeled Self Balancing Autonomous Mobile Robot

Victoria Gutiérrez-Vicente, Juan Díaz-Téllez,
Jaime Estevez-Carreon, Jairo Pérez-Pérez

Tecnológico Nacional de México,
Instituto Tecnológico de Puebla,
Mexico.

{juan.diaz, jaime.estevez, jairo.perez ,
i17221010.09}@puebla.tecnm.mx

Abstract. This paper presents the control of the position of a Segway robot through a feedback of the state vector. We demonstrate that using this simple linear approach the Segway personal transportation can be stabilized. The approximate linearization is used in the nonlinear model for the design of the control law, advantages such as the easy implementation in an embedded system and the low calculation cost in processors has motivated choosing this approach. To validate the design of the control law, three parameters have been taken into account: settlement time, ability to reject external disturbances, and trajectory tracking.

Keywords. Segway, linear control, ESP32, robot operating system (ROS), full state feedback control.

1 Introduction

Mobile robotics is one of the sciences that has received attention in the industrial-academic and social fields. This is due in large part to the development of embedded computing systems, microelectromechanical systems MEMS sensors, devices with high energy storage and operating systems. Mobile robots have a wide niche of applications: tracking and transfer of objects, evasion of obstacles, cooperative environments, analysis and inspection, exploration of remote areas to mention some. In particular, land mobile robots are the research objective in this work.

These robots can be divided mainly by the mechanisms they use to move, generally divide them into wheels, tracks and legs. The simplest case of mobile robots is wheeled robots. Wheeled robots comprise one or more driven wheels and have optional passive or caster wheels and possibly steered wheels. The arrangement of the driven, passive and steered wheels gives different configurations of land mobile robots.

Today, balancing robots have gained popularity with the introduction of the commercial Segway vehicle. These robots have the ability to balance on their two wheels, making them excellent personal electric transport. Due to their great maneuverability, these robots allow people to be transported over short distances with

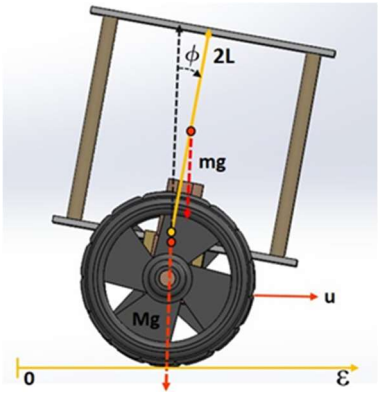


Fig. 1. Two-wheeled self-balancing.

a moderate speed, navigate over confined spaces indoors and outdoors, turn on their axis and in sharp corners, in addition to going through small unevennesses.

The physical balancing robot is an inverted pendulum with two independently driven motors, to allow for balancing, as well as driving straight and turning [1].

The balancing robot has two independent fixed motors to drive the wheels at each end, unlike the differential robots, the balancing robot lacks passive wheels, also the bar and the person on board makes the stabilization of the system more complicated.

In [3] linear LQR control is designed, it is implemented on an embedded FPGA-based system. The results show indoor experimental tests mounting stable robot control. In [4] a linear LQR control and an ISMC are designed, the mathematical model is based on the inverted pendulum on a mobile platform, the comparison between both algorithms is performed in simulation. In [5] a control based on a geometric PID is designed. This approach ensures almost global locally exponential stability of the upright motion of the Segway. The effectiveness of the control law is demonstrated through simulations against uncertainties.

In [6] a Segway with multidirectional shock absorbers between the vehicle motors and the main user platform are designed. The control scheme implemented is a PID controller when the vehicle is moving. Also, when the Segway is stationary, the PID controller ensures the smooth balancing of the vehicle. The state equations found from the mathematical modelling of the vehicle were used in MATLAB to calculate the optimum compensation constant values for the controller.

On the other hand, in [7] an optimal control method for the nonlinear system of Segway PT is proposed for reduces energy consumption and enhance the response speed of the system instead of classic PID controller. The controller showed the balance between speed response and control cost and the rejection of disturbance. Finally, in [8] a linear LQR control is implemented to control the Segway by adjusting parameters of controller gain. The results focus on the control algorithm which can adapt controller gain by driver weight.

The paper is structured as follows. In section 2 some mathematical preliminaries used along the document are presented, also the presentation of the model of the Segway Robot. In section 3 presents the control law design based on state feedback. In

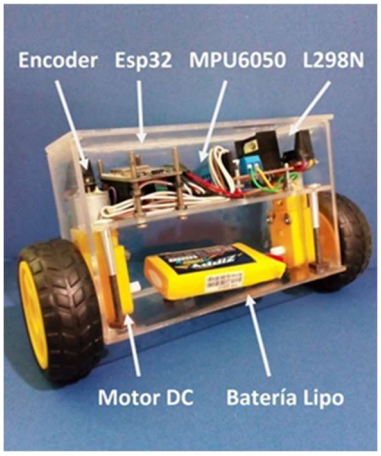


Fig. 2. Low-cost prototype of the inverted pendulum robot.

the 4 the simulation results are present which show the effectiveness of the proposed algorithm. In section 5 some conclusions and future work are presented.

2 Two-Wheeled Self-Balancing System Model

The segway robot model that we present is based on the inverted pendulum principle. The dynamics of the inverted pendulum has been the basis for bipedal robots, primary space propellers, and attitude control of small satellites.

The basic movements of the robot, driving forward-backward movement, driving sideways movement and rotate on its own axis.The forward-backward movement is achieved when the speed of both motors are increased or decreased by the same amount.

The sideways movement is achieved when the speed of the left motor is increased, while the speed of the right is decreased and vice versa. The rotate on its own axis is achieved when the speed of both motors are increased in the opposite direction.

The embedded system is composed of a 32-bit dual core ESP32 microcontroller with a 240 MHz clock frequency. This chip contains wireless communication modules such as WiFi and Bluetooth. The actuation system is made up of two - 5 V permanent magnet direct current motors.

The power system has a 1000 mAh 2s Zippy Lipo Battery, an MPU-6050 inertial unit and a L298N H-bridge as the power stage. Selecting the following variables (ξ, φ) and using the Euler-Lagrange formalism, the dynamics that describe the physics of the vehicle is obtained. The nonlinear model of the inverted pendulum is can be modeled as:

$$(m+M)\ddot{x} + (ml\cos\theta)\ddot{\theta} = -C_1\dot{x} + (ml\sin\theta)\ddot{\theta}^2 + u, \quad (1)$$

$$(ml\cos\theta)\ddot{x} + (J+ml^2)\ddot{\theta} = -C_2\dot{\theta} + (mgl\sin\theta), \quad (2)$$

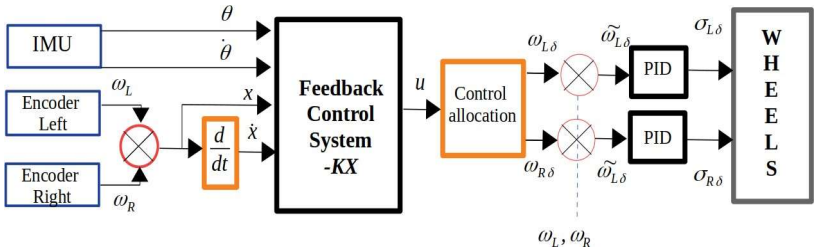


Fig. 3. Block diagram of the Segway control system.

where the state variables are defined as the displacement of the mobile x, the speed of the mobile as $\dot{x}$, as well as the angular displacement of the bar and the rider θ and its angular velocity $\dot{\theta}$.

The variable u represents the force that pushes the system in order to control it, therefore they are the signals from the motors.The constants J is the moment of inertia with respect to the center of gravity ofthe pendulum, l represents the length of the bar to the center of mass, M is the mass of the base of the Segway, m is the mass of the bar and the human, and grepresents gravity. The constants C_1 and C_2 represent the coefficients of frictionof the rotary movement of the pendulum and of the linear movement of the base.For the mathematical control model, some restrictions have been considered:

- The center of gravity of the bar is at its geometric center.
- The coefficients of friction of the rotary movement of the pendulum and the linear movement of the base are negligible.
- The angular displacement θ is very small.

The Segway model for the design of the control algorithm is:

$$\ddot{x} = \frac{-m^2 l^2 g\theta + (J + ml^2)u}{\Delta}, \tag{3}$$

$$\ddot{\theta} = \frac{(M+m)(mgl\theta) - mlu}{\Delta}, \tag{4}$$

where $\Delta = (M+m)(J+ml^2) - m^2l^2$. Selecting as state variable $x_1 = x$, $x_2 = \dot{x}, x_3 = \theta$ and $x_4 = \dot{\theta}$. Consider the equilibrium point of this system in the upright vertical position:

$$[x_1\ x_2\ x_3\ x_4]^T = [x_d\ 0\ 0\ 0]^T \text{ further } u = 0, \tag{5}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{(J+ml^2)u - m^2l^2gx_3}{\Delta} \\ x_4 \\ \frac{-mlu + (M+m)mglx_3}{\Delta} \end{bmatrix}, \tag{6}$$

The approximate linearization of the system at the equilibrium point is:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{-m^2 l^2 g}{\Delta} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)mlg}{\Delta} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{J+ml^2}{\Delta} \\ 0 \\ \frac{-ml}{\Delta} \end{bmatrix} u. \tag{7}$$

For the mathematical control model, some restrictions have been considered:

- The center of gravity of the bar is at its geometric center.
- The coefficients of friction of the rotary movement of the pendulum and thelinear movement of the base are negligible.

2.1 Propulsor Model

The system includes a brushless DC motor, ESC and propellers. First, givena constant voltage by the LIPO battery, the ESC generate an average voltage signal which is a function of the pulse width modulation PWM signal which will make the motor achieve steady-state speed. The above can be modeled as:

$$w_{ss} = C_R \sigma + w_b, \tag{8}$$

where C_R is a constant that depends on the battery voltage and the ESC, σ is PWM signal with a value between 1000 and 2000 indicate the duty cycles, and w_b is the angular speed which the motor reaches once the motor initializes. Because the motor needs some time to achieve the steady state speed Ω_{ss}, a second equation is generated to model the dynamic response. Generally, the dynamics of a brushless DC motor can be simplified as a first order equation:

$$w = \frac{1}{T_m s+1} w_{ss}, \tag{9}$$

Combining the Equations 9 and 8 get the complete propulsor model:

$$w = \frac{1}{T_m s + 1}(C_R \sigma + w_b \,. \tag{10}$$

3 Full State Feedback Control

The strategy to follow is to design a linear control law based on the state feedback for the lineal system eq. (7), so that it stabilizes the state variables to zero. Once the linear system has stabilized, the linear control law is applied tothe nonlinear system eq. (1) and eq. (2). This would ensure that the system is operating at its equilibrium point.

We verify that the linear system is controllable:

$$\det(C) = \det([B : AB : A^2B : A^3B]) \neq 0, \tag{11}$$

We propose the following control law:

$$u = -k_1 x_1 - k_2 x_2 - k_3 x_3 - k_4 x_4, \tag{12}$$

Propose that the linearized closed-loop system have its poles located at the roots of the following desired polynomial:

$$p_d(s) = (s + \alpha)^2(s^2 + 2\zeta w_n s + w_n^2). \quad (13)$$

where ζ is the damping factor and w_n is the undamped natural frequency corresponding to the pair of complex conjugated poles generating the second degree factor. Solving the system of equations:

2.2 Control Motor

Efficient BLDC motor speed control is required for the Segway robot. A quick and smooth response will produce high performance in position control.We assume that the angular speed of the brushless direct current motor is w, furthermore these motors use high frequency pulse width modulation (PWM) to control the motor voltage. The control objective is to design each motor a PWM signal command, such that $\lim_{t\to\infty}|w_d(t) - w(t)| = 0$. We propose a PID control to minimize the error signal $\breve{w} = w_d(t) - w(t)$:

$$\sigma = k_p\breve{w} + k_i \int \breve{w} + k_d\breve{w}. \quad (14)$$

The angular speed of the motors w can be obtained from the encoder, C_R and w_b are obtained through experimentation.

4 Simulation

The figure 3 shows the block diagram of the system. The system has two main sensors, an Inertial Measurement Unit (IMU) and an encoder on each BLDC motor. The IMU sensor estimates the angle θ using a built-in Kalman filter and the angular velocity $\dot{\theta}$. The encoders provide information on the angular speed of the wheel, with the angular speed we can calculate the position x and the speed of the base $\dot{x}$. The four state variables are fed back to the control algorithm and it is calculated u.

The control allocation block allows the control signal to be converted into the required angular velocity $w_{L\delta}$ and $w_{L\delta}$ for the motors. Lastly, a PID is necessary to ensure the desired angular velocity. The choiceof w_n and ζ is made to set the rise time of the system.

Typically ζ is chosen as 0.707, in order to achieve a damped response, w_n is chosen as $12 = {}^{rad}/_{S}$. The figures show the evolution of state variables of the non-linear system controlled by the feedback law based on approximate linearization in different conditions.The parameters for the simulation of the system are $M = 0.43,\ m = 0.16, l = 0.25, J = 0.0043, g = 9.81$ The vector values $K = [k_1 k_2 k_3 k_4]^T$, is obtained by solving the system of equations that is obtained by equating the coefficients of the desired polynomial term by term, k_1=-195.71,k_2=-78.28,k_3=-192.42,k_4=-36.05. In he figures 4, 5, 6 and 7 we try to reproduce the forward-backward command with the inclination of the bar by θ=23^o, that is to say that propose X_δ=[0 0 20 0]^T as equilibrium point for the

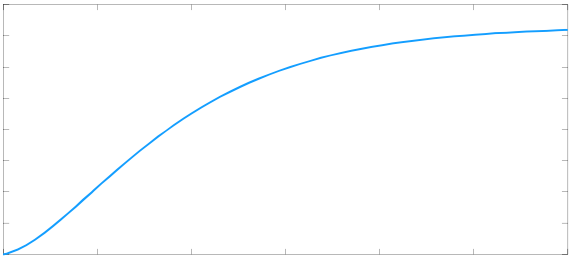


Fig. 4. Forward-backward movement, x evolution.

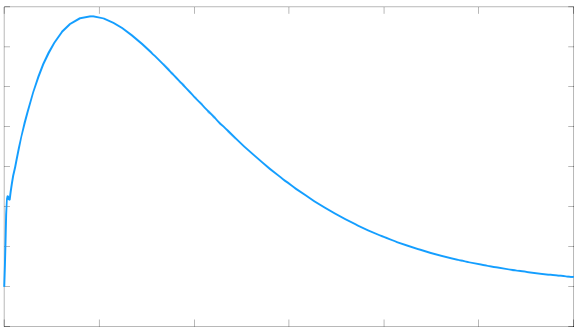


Fig. 5. Forward-backward movement, $\dot{x}$ evolution.

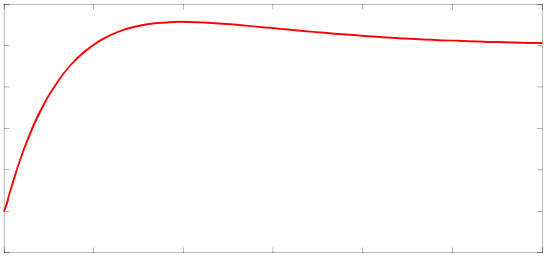


Fig. 6. Forward-backward movement, θ evolution.

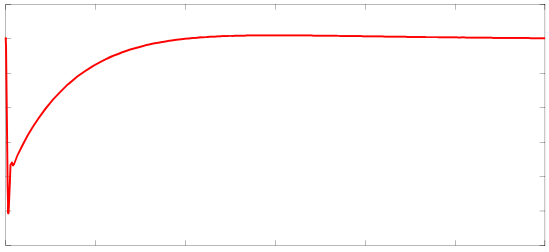


Fig. 7. Forward-backward movement, $\dot{\theta}$ evolution.

following initial conditions x(0)=[0 0 0 0]^T. As we can see in the images when tilting the bar by an angle θ, the base tries to compensate the bar to its vertical position, therefore its movement is towards forward.

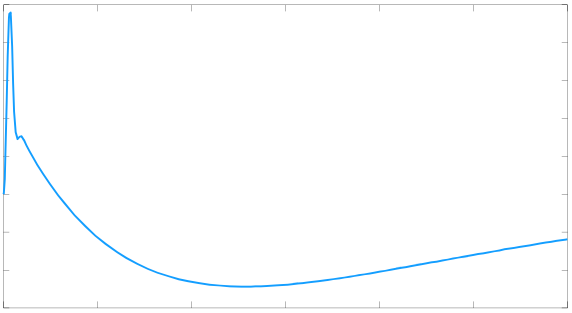


Fig. 8. Hover movement, x evolution.

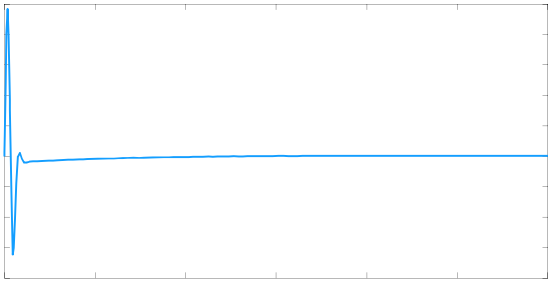


Fig. 9. Hover movement, $\dot{x}$ evolution.

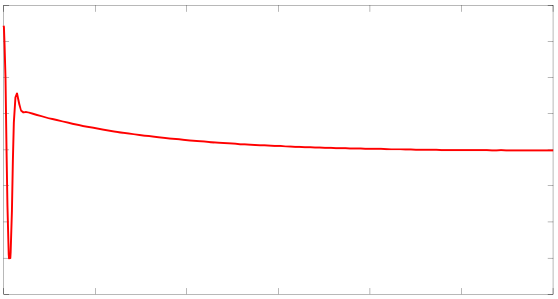


Fig. 10. Hover movement, θ evolution.

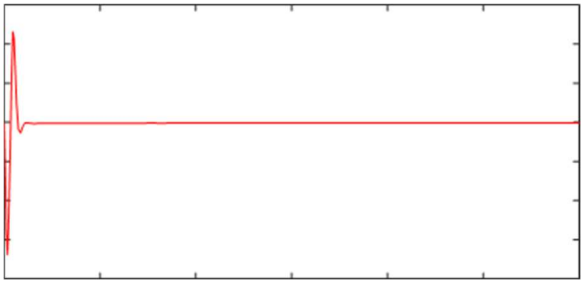


Fig. 11. Hover movement, $\dot{\theta}$ evolution.

The figures 8, 9, 10 and 11 show a hover movement while the robot bar is disturbed. The initial system condition is $x(0) = [0\ 0\ 35\ 0]^T$.

The figure 10 shows the position of the bar with an initial condition at 35^o can see how the bar stabilizes at zero due to the movement of the base see figure 8. Stabilization time is approximately 2.5 seconds, which is an excellent time for these systems.

5 Conclusions

It should be emphasized that the linearized system-based design applied to the nonlinear system at least around its equilibrium point may not stabilize the system if the disturbances move away from the equilibrium point, however, due to the nature of the Segway robot (small angles) the linear control law proposed by state feedback is stable enough for the stability of said robot. The design, instrumentation and implementation of the embedded system will represent a great challenge, an ARM microcontroller with an SBC is proposed for the embedded control system.

References

1. Braunl, T.: Embedded robotics: Mobile robot design and applications with embedded systems: Second edition (2006) doi: 10.1007/3-540-34319-9
2. Sira-Ramirez, H. T., Marquez, R., Rivas, F., Orestes, S.: Control de sistemas no lineales: Linealización aproximada, extendida (2018)
3. Pinto, L. J., Kim, D., Lee, J. Y., Han, C.: Development of a segway robot for an intelligent transport system. In: IEEE/SICE International Symposium on System Integration (SII) (2012)
4. Shilpa, B., Indu, V., Rajasree, S. R.: Design of an underactuated self-balancing robot using linear quadratic regulator and integral sliding mode controller. In: International Conference on Circuit, Power and Computing Technologies (ICCPCT) (2017)
5. Basnayake, I. D., Madhushani T. W. U., Maithripala, D. H. S.: Intrinsic PID controller for a segway type mobile robot. In: IEEE International Conference on Industrial and Information Systems (ICIIS) (2017)
6. Draz, M. U., Ali, M. S., Majeed, M., Ejaz, U., Izhar, U.: Segway electric vehicle human transporter. In: IEEE International Conference of Robotics and Artificial Intelligence (ICRAI) (2012)
7. Babazadeh, R., Khiabani, A. G.: Optimal control of segway personal transporter. In: IEEE International Conference on Control, Instrumentation, and Automation (ICCIA) (2016)
8. Chantarachit, S.: Development and control segway by LQR adjustable Gain. In: IEEE International Conference on Information and Communications Technology (ICOIACT) (2019)

ISSN 1870-4069

A First Approach to Food Composition Estimation as an Image Classification Problem

Jorge Alberto Cabrero-Dávila, Humberto Pineda-Ivo

Benemérita Universidad Autónoma de Puebla,
Departamento de Ciencias Computacionales,
México

{jorgedavila33, ivopinedatorres}@gmail.com

Summary. Food related degenerative diseases levels have increased dramatically in the last decades. Food and diet guidelines promoted by governments and health agencies haven't had the desired effect on reversing these trends. One of the principal explanations is that guidelines are hard to follow and are unclear. Deep Learning algorithms can help to make easier for people to follow dietary guidelines using technology available to them. In this work, a first approach is proposed to solve an early stage of the entire scope of the problem which is the estimation of calories and nutrients in food based on their image.

Keywords: Food classification, image classification, deep learning.

1 Introduction

Calorie counting is one of the two major components in almost all issued health guidelines. Research has shown that restricting calories helps reduce weight and increase health markers such as blood pressure, insulin resistance, cholesterol levels, liver health, etc. [1]. The other major component is nutrient density of food, it is well known that the body needs certain nutrients (vitamins, minerals, and fiber) and macro-nutrients (proteins, carbs, and fats) to be healthy and that these nutrients and macro-nutrients are only available in food and can't be obtain elsewhere [2].

Following the major components of guidelines; a diet, is composed by a recommended number of calories based on age, gender, current weight, and activity. Also, by several levels of micro and macro nutrients, measured in grams, milligrams or micro-grams depending on the nutrient.

For example, a 25-year-old male with average activity of 30 minutes a day could need a diet of 2500 calories, 1 gram of protein per kilo of body weight, 30 grams of carbs, 10 milligrams of zinc, 30 milligrams of manganese, etc.

With that target in mind there are countless ways to achieve those levels and it is very easy to overpass those limits calorie-wise and be in the low side nutrient-wise. To stick to the recommended, the person should first know the composition of a lot of food,

Table 1. Datasets used.

	Images	Classes	Original Size	Size after treatment
Food 101	101,000	101	512 x 384	224 x 224
ECUSTD	2,987	20	816 x 551	224 x 224

Fig. 1. ECUSTD Data base examples. From left to right from top to bottom: apple, banana, bread, bun, donut, egg, lemon, mango, pear, kiwi, plum, sachiman, orange, mix. Source: compiled by the authors based on ECUSTD Database.

Fig. 2. Food 101 Data base examples. From left to right from top to bottom: baklava, beef tartare, Caesar salad, ceviche, deviled eggs, filet mignon, fried rice, hamburger, pizza. Source: compiled by the authors based on Food 101 Database.

and secondly, they should know exactly the amount they're eating, both aspects are difficult and impractical in a modern lifestyle.

If only there was a way to easily recognize, access and measure food composition without weight it or search for it would mean a major step for achieving better public health around the globe. In this work we believe that computer vision paired with deep learning algorithms can achieve just that by just taking a photo with a cell phone or the like of the food that it's about to be eaten.

2 State of the Art

Currently there aren't many works on the subject in question; however, tech giant Google have already started to research and develop in this matter with its leading scientist Kevin Murphy. They already have developed Im2calories an app that can count calories from a low-resolution photo taken with a cellphone of a dish [3], it only works in 30 percent of the cases, but they are improving it with people inputs since 2015. The model is not open to the public but is based on a convolutional neural network.

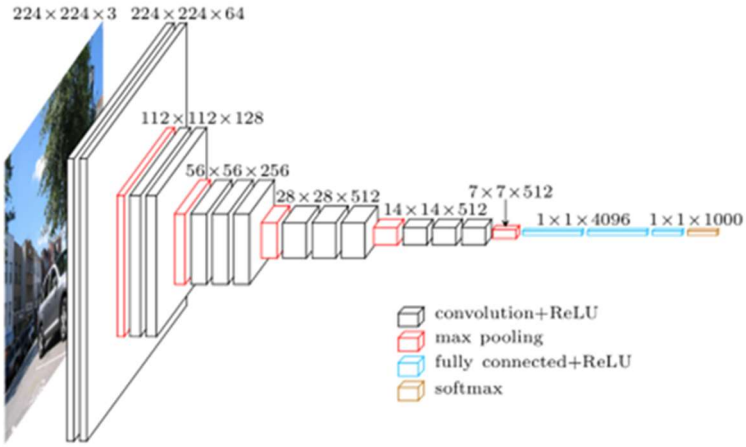


Fig 3. VGG16 Architecture [11].

Table 2. GoogLeNet Architecture [10].

Operator	Input
Conv1	3 x 224 x 224
Pool1	64 x 112 x 112
Conv2	64 x 56 x 56
Pool2	192 x 56 x 56
Inception3a	192 x 28 x 28
Inception3b	256 x 28 x 28
Pool3	480 x 28 x 28
Inception4a	480 x 14 x 14
Inception4b	512 x 14 x 14
Inception4c	512 x 14 x 14
Inception4d	512 x 14 x 14
Inception5e	512 x 14 x 14
Pool4	832 x 14 x 14
Inception5a	832 x 7 x 7
Inception5b	832 x 7 x 7
Pool5	1024 x 7 x 7
Fc	1024 x 1 x 1
prob	1000

Yanchao Liang and Jianhua Li from the East China University of Science and Technology in 2017 [4], created the ECUSTD database composed of 2987 images of 20 categories of food, also they introduced a CNN architecture for image classification and segmentation of food to estimate food calories. In their work results varied greatly, they predicted the weight of the food following a process of object recognition, image segmentation and image classification.

The best accuracies were for simple foods such as an orange with a mean error of 0.7%, plum with -2.1% and lemon with mean error of -2.8%. Other simple foods such

as apple, grape and banana had large mean errors: -18.7%, 33.5% and 28.8% respectively.

Simon Mezgec and Barbara Seljak from the Information and Communication Technologies department of the University of Slovenia developed in 2017 the NutriNet a CNN architecture that they claim to have 94 percent accuracy in some food datasets (classification only) [5].

Xioweng Wu et. al. from the Living Analytics Research Center of the University of Singapore, compiled a database (FoodSeg103) of 9490 images with 154 classes and segmentation masks [6].

Additionally, they proposed a multimodal pre-training approach called ReLeM which equips the model with semantic knowledge. They performed three segmentation experiments: Dilated Convolution based, Feature Pyramid based, and Vision Transformed based. The base and the models they proposed are intended to function as a point of reference for future work.

We can conclude from the works cited above and some others not mentioned that the problem of food recognition is challenging due to the nature of food items. Foods are deformable objects, which makes the process of defining their structure difficult. Furthermore, some foods types can have high intra-class and low inter-class variance. It seems that a deep learning approach can be very promising in the field of food image recognition.

None of the works reviewed so far have tried to estimate nutritional value on top of calorie estimation so it seems that there is a lot of room for improvement and the problem cannot be considered solved yet.

3 Methodology

The objective of this work is to develop a first approach for the complex problem of food composition estimation based on computer vision and deep learning. The problem is complex, and it requires two main treatments that need to be combined. The first one is pure image classification [7] (classifying foods based on the entire image they appear on), the second is object recognition and segmentation (segmenting different food items in the same image) [8].

In this work we focus on image classification only, we believe that the correct identification of food type is the first step for an accurate food composition estimation later.

Given the promising results of other works and the fact that convolutional neural networks are used to solve complex problems [9], we decided to test well known CNN's architectures in food data sets in order to get a grasp on food image classification.

We mainly used two datasets: The Food 101 dataset formed by 101,000 images of 101 food classes gather by Bossard et al. for their work in Mining Discriminative components with random forests. The dishes presented in this dataset can be as simple as an apple pie or complex as a Caesar salad.

Table 3. MobileNet Architecture [12].

Operator	Input
Conv / s2	3 x 224 x 224
Conv dw / s1	32 x 112 x 112
Conv / s1	32 x 112 x 112
Conv dw / s2	64 x 112 x 112
Conv / s1	64 x 56 x 56
Conv dw / s1	128 x 56 x 56
Conv / s1	128 x 56 x 56
Conv dw / s2	128 x 56 x 56
Conv / s1	128 x 28 x 28
Conv dw / s1	256 x 28 x 28
Conv / s1	256 x 28 x 28
Conv dw / s2	256 x 28 x 28
Conv / s1	256 x 14 x 14
5 x (Conv dw /s1, Conv / s1)	512 x 14 x 14
Conv dw / s2	512 x 14 x 14
Conv / s1	512 x 7 x 7
Conv dw / s2	1024 x 7 x 7
Conv / s1	1024 x 7 x 7
Avg Pool / s1	1024 x 7 x 7
FC / s1	1024 x 1 x 1
Softmax / s1	1000

The other dataset we used is the ECUSTD dataset gather by Yanchao Liang and Jianhua Li from the East China University of Science and Technology. This dataset is formed by 2987 images of 20 classes of food, the classes are simple as apple, banana, grapes, mango, doughnut, etc. There is only one class labeled as mix which include two foods from two different classes. Images in both datasets were treated to fit the size 224 x 224 pixels which is the size used in the most popular CNN architectures built for the ImageNet Challenge.

The treatment used to resize the images was a bi-cubic interpolation over a 4x4 pixel neighborhood. The language used was Python with cv2 library. We choose 3 architectures to test food classification: MobileNet, GoogLeNet and VGG16. The decision behind testing GoogLeNet seems obvious; the model has the lowest count of parameters of the modern architectures paired with great accuracy obtained in the ImageNet challenge. On the other hand, we choose the VGG16 model due to the simplicity and elegance of the architecture combined with its low error.

AlexNet and ZFNet were left aside due to the high number of parameters in the model together with a higher error-rate. ResNet model was left aside too, this time the decision was since the model has 152 layers, and the size of the images is difficult to fit to the model therefore we should have used the ResNet18 model instead which has lower accuracy than GoogLeNet and VGG16.

GoogLeNet and VGG16 were the 1st and 2nd places respectively in the ImageNet Large Scale Visual Recognition Challenge in 2014. GoogLeNet has 22 layers and VGG 16 has 19 layers. Both architectures are big and slow to train, but fast to use in

Table 4. Comparison of ImageNet champions and runner ups. Source: Compiled by the authors based on the results of the ImageNet Challenge.

Year	CNN	Developed by	Place	Top-5 Error Rate	No. of parameters
1998	LeNet	Yann Le Cunn et al			60 thousand
2012	AlexNet	Alex Krizhenvsky et al.	1st	15.3%	60 million
2013	ZFNet	Matthew Zeiler et al	1st	14.8%	60 million
2014	GoogLeNet	Google	1st	6.67%	4 million
2014	VGG16	Simonyan and Zisserman	2nd	7.3%	138 million
2015	ResNet	Kaiming He	1st	3.6%	60 million

predictions once the training is over. Table 4 shows a comparison of several champions of the ImageNet challenge and their characteristics.

MobileNet was also chosen to test the accuracy of the classification task. The MobileNet it's part of a family of MobileNets designed also by Google to effectively maximize accuracy while being mindful of the restricted resources for an on-device or embedded application.

MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. They can be built upon for classification, detection, embeddings, and segmentation.

MobileNet architecture uses depth-wise convolutions (Conv dw), where each filter channel is used only at one input channel.

We made some changes in the fully connected layers of all three architectures before training them with the food images datasets. For VGG16 we changed the number of neurons in the fully connected layers, originally the architecture has 3 fully connected layers of 4096 neurons each and a final output of 1000. We changed the number of neurons to 512 for the 3 final layers and the final output was changed to 20 for the ECUSTD training and to 101 for the Food 101 training.

For the GoogLeNet and MobileNet we changed the number of neurons in the final output layer in order to match the number of categories in both databases. The 3 architectures were trained on both datasets, for the ECUSTD dataset we used two techniques for the training: transfer learning and full learning.

Given that the Food 101 data set is considerably larger it was only tested with transfer learning. The transfer learning technique applied in this work was to freeze the convolutional layers of the network with the weights trained with the ImageNet dataset and only train the fully connected layers for 100 epochs. The full training technique trains all layers including the convolution layers with the selected dataset for 100 epochs.

Table 5. Results – ECUSTD (in Colab).

Model	Transfer Learning	Epochs	Optimizer	Validation	Acc.	Time
GoogLeNet	True	100	Adam	5 folds	99.56	1 h
GoogLeNet	True	100	SGD	5 folds	99.43	45 m
GoogLeNet	False	100	Adam	5 folds	99.6	1.7 h
VGG16	True	100	Adam	5 folds	10.4	1.25 h
VGG16	False	100	Adam	5 folds	5.01	2 h
MobileNet	True	100	Adam	5 folds	99.76	40 m
MobileNet	False	100	Adam	5 folds	99.63	1.5 h

Table 6. Results – Food 101 (in Buap Ser).

Model	Transf Learning	Epochs	Optimizer	Validation	Acc.	Time
GoogL Net	True	100	Adam	5 folds	63.08	12 h
VGG16	True	100	Adam	5 folds	18.14	24 h
Mobileet	True	100	Adam	5 folds	66.3	14 h

The Loss function used in all experiments was Cross Entropy Loss Function, the Adam Step optimizer was use in most of the experiments except for one where Stochastic Gradient Descent was used. The validation method used was a 5-fold cross validation.The experiments were carried on using python's library Pytorch and run in Google's colaboratory and BUAP's server.

4 Results

Tables 5 and 6 shows the results of the different experiments performed. From the results in tables 5 and 6 it can be seen that for the ECUSTD dataset a very good accuracy can be obtained with transfer learning in two of the three architectures with relative low training times. VGG16 had the lowest accuracy with both transfer learning and without it, it's important to point out that VGG16 is the model with more parameters to optimize and that's why the training times are the largest in general.

MobileNet and GoogLeNet didn't improved the accuracy a lot by making a full training without transfer learning. For the Food 101 data set the highest accuracy achieved was obtained with GoogLeNet, second and very close was the MobileNet. VGG16 as with the other dataset failed to do a decent classification.

We can conclude that food classification using transfer learning and well-known CNN architectures can be satisfactory achieved in cases where the foods classes are simple and isolated in an image, for example, images of fruits.

In the other hand, when complex dishes are taken into consideration the recognition drops dramatically, however, a 60% accuracy for complex dishes without any kind of prior segmentation or object detection reflects a positive result, because it means that with proper segmentation and detection algorithms, accuracy can improve a lot.

5 Conclusions

Further experiments need to be carried out with the VGG16 architecture, because the default experiments achieved a very low accuracy, changes in the fully connected

layers, epochs, optimizer, and kernel sizes can be tested in order to see if the accuracy can increase. The results of the experiments confirm our assumption that this approach can be a first step for a full food composition process.

Next steps could include object detection and segmentation algorithms paired with the image classification methods mentioned above. Furthermore, even though a lot of improvements have been done in Deep Learning with CNN algorithms, for food composition estimation exists a real challenge to construct a model that can classify food correctly, one of the major setbacks is that there are not many datasets on the subject, many of them are small or they represent small classes of food.

Also, many food dishes have irregular shapes and wide color ranges, and that represents a very complex problem to solve even today with the most robust of architectures. Future work can focus too on obtaining datasets of regional foods, as we believe also that regional foods images can increase the accuracy in any model trying to classify dishes of a given country diet.

References

1. Varady, K. A.: Intermittent versus daily calorie restriction: which diet regimen is more effective for weight loss? Obesity reviews, vol. 12, no. 7, pp. 593–601 (2011)
2. Sauberlich, H. E.: Bioavailability of vitamins. Progress in food and nutrition science, vol. 9, no. 1–2, pp.1–33 (1985)
3. Meyers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Murphy, K. P.: Im2calories: Towards an automated mobile vision food diary. In: Proceedings of the IEEE International Conference on Computer Vision pp. 1233–1241 (2015)
4. Liang, Y., Li, J.: Computer vision-based food calorie estimation: Dataset, method, and experiment (2017) arXiv preprint arXiv:1705.07632
5. Mezgec, S., Koroušić Seljak, B.: NutriNet: A deep learning food and drink image recognition system for dietary assessment. Nutrients, vol. 9, no. 7, pp.657 (2017)
6. Wu, X., Fu, X., Liu, Y., Lim, E. P., Hoi, S. C., Sun, Q.: A large-scale benchmark for food image segmentation. pp. 506–515 (2021) doi: 10.1145/3474085.3 475201
7. Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25 (NIPS 2012)
8. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs (2014) doi: 10.48550/arXiv.1412.7062
9. Patel, S.: A Comprehensive analysis of convolutional neural network models. In: Proceedings of International Journal of Advanced Science and Technology, vol. 29, no. 4. pp. 771–777 (2020)
10. Kölsch, A., Afzal, M. Z., Liwicki, M.: Multilevel context representation for improving object recognition. In: Proceedings of 14th IAPR International Conference on Document Analysis and Recognition, pp. 10–15 (2017) doi: 10.1 109/ICDAR.2017.322
11. Qassim, H., Verma, A., Feinzimer, D.: Compressed residual-VGG16 CNN model for big data places image recognition. In: IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), pp. 169–175 (2018)
12. Qiya, N., Yunlai, T., Lin, C.: Design of gesture recognition system based on deep learning. In: Proceeding of Journal of Physics: Conference Series, vol. 1168, no. 3 (2019) doi: 10.1088/1742-6596/1168/3/032082

ISSN 1870-4069

Localización y reconocimiento de matrículas de automóviles utilizando redes neuronales

Elizabeth Xicotencatl-Flores, Aldrin Barreto-Flores,
Salvador E. Ayala-Raggi, Verónica E. Bautista-López

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Electrónica,
México

elizabeth.xicotencatlf@alumno.buap.mx, {aldrin.barreto,
salvador.raggi, veronica.bautistalo}@correo.buap.mx

Resumen. Este artículo propone un método de reconocimiento de placas vehiculares usando el modelo de intersección cortical con conexiones nulas (ICMNI) para la localización de los caracteres y un perceptrón multicapa (MLP) de una sola capa oculta para su clasificación por separado. En contraste a otros trabajos reportados, este algoritmo excluye el proceso de localización de la placa y el preprocesamiento de la imagen permitiendo reducir el tiempo de procesamiento, además de aumentar la tasa de reconocimiento en imágenes donde los niveles de intensidad no son homogéneos. Para el entrenamiento se creó un dataset con 6,600 caracteres considerando únicamente 10 números y 23 letras. Se llevaron a cabo pruebas del algoritmo propuesto en un conjunto de 660 imágenes traseras y frontales de automóviles obtenidas de diversos sitios web compatibles con el formato asignado al estado de Puebla generando como resultado una tasa de reconocimiento del 96.2%.

Palabras clave: Mexican license plate recognition, intersecting cortical model, pulse coupled neural network, multilayer perceptron.

Localization and Recognition of Car License Plates Using Neural Networks

Abstract. This article proposes a vehicle license plate recognition method using the null-connected cortical intersection model (ICMNI) for character localization and a single hidden layer multilayer perceptron (MLP) for their separate classification. In contrast to other reported works, this algorithm excludes the plaque localization process and the image preprocessing, allowing to reduce the processing time, as well as increasing the recognition rate in images where the intensity levels are not homogeneous. For training, a dataset with 6,600 characters was created considering only 10 numbers and 23 letters. Tests of the proposed algorithm were carried out on a set of 660 rear and front images of automobiles obtained from various websites compatible with the format assigned to the state of Puebla, generating as a result a recognition rate of 96.2%.

Keywords: Mexican license plate recognition, intersecting cortical model, pulse coupled neural network, multilayer perceptron.

1. Introducción

Los Sistemas de reconocimiento de matrículas han sido estudiados a nivel mundial a través de múltiples aplicaciones (que van desde el cobro automatizado de estacionamientos [1] hasta sistemas de vigilancia [2]), cada uno con sus propias restricciones por las necesidades a cubrir en su diseño contemplando factores como: cambios de iluminación, resolución y posición de la cámara, oclusión, deterioro de las matrículas y el más importante, la tipografía de cada país. Este es el motivo de que cada metodología sea independiente y se limite a ciertas condiciones y fondos para evitar un bajo desempeño en imágenes con altas variaciones.

Actualmente es posible identificar dos enfoques en este tipo de sistemas, los de una sola etapa y los multietapa [3]. Los primeros se basan fundamentalmente en algoritmos de aprendizaje profundo (también conocido como *deep learning*) conformados por redes neuronales artificiales que emplean la detección de objetos en extensas bases de datos donde se requiere ciertos niveles de abstracción para localizar y clasificar las imágenes.

Entre las redes pre-entrenadas más populares se encuentran las convolucionales como *Single-Shot Detector* (SSD), *You Only Look Once* (YOLO), *Region Based Convolutional Neural Networks* (R-CNN) y sus versiones optimizadas [4-7]. En el segundo enfoque se crean tres subprocesos: localización de la placa, segmentación y reconocimiento de los caracteres, los cuales involucran diferentes técnicas de procesamiento de imágenes y clasificadores tales como redes probabilísticas [8], correlación (*template matching*) [9], *K-Means* [10], y *k-nearest neighbors* (k-NN) [11]. Por lo general, la elección de cada técnica depende de los recursos disponibles y de la tasa final de reconocimiento que cada subproceso añade a la etapa anterior.

En este trabajo se presenta un algoritmo para el reconocimiento de matrículas de automóviles basado en el modelo de intersección cortical con conexiones nulas para el proceso de segmentación cuya efectividad recae en la condición de activación continua de cada neurona para la asignación de los parámetros y un perceptrón multicapa de una sola capa oculta para la clasificación de los caracteres.

Este artículo está organizado de la siguiente manera: en la sección 2 se presentan los fundamentos de las técnicas empleadas, en la sección 3 se muestra el método propuesto a detalle para el reconocimiento de placas vehiculares, en la sección 4 se muestran los resultados experimentales y su comparación con otras técnicas actuales y finalmente las conclusiones se mencionan en la sección 5.

2. Marco teórico

2.1. ICM con conexiones nulas

Una red pulso acoplada (PCNN, *Pulse Coupled Neural Network*) es un tipo de red neuronal que clasifica los píxeles por sus niveles de intensidad generando una secuencia de imágenes binarias a partir de una sola imagen de entrada sin necesidad de una etapa de entrenamiento, razón por la cual ha sido aplicada a la segmentación, codificación, fusión y extracción de características de imágenes [12-15], siendo una técnica poco explorada para la segmentación de matrículas vehiculares.

El modelo de intersección cortical (ICM, *Intersecting Cortical Model*) es una variante de una PCNN diseñada para reducir el costo computacional eliminando algunos parámetros de ajuste del modelo original. En cada iteración la intensidad de los píxeles que conforman la imagen a procesar es introducida a la red a través de una señal conocida como *Feeding F*, la cual se compara con un umbral dinámico *E* para generar los pulsos de salida *Y* representados por *n* imágenes binarias:

$$F_{ij}(n) = f F_{ij}(n-1) + S_{ij} + \sum kl\, W_{ijkl} Y_{kl}(n-1), \tag{1}$$

$$E_{ij}(n) = gE_{ij}(n-1) + hY_{ij}(n-1), \tag{2}$$

$$Y_{ij}(n) = \begin{cases} 1, & F_{ij}(n) > E_{ij}(n), \\ 0, & en\ caso\ contrario. \end{cases} \tag{3}$$

El comportamiento anteriormente definido puede ser descrito por las ecuaciones (1)-(3), donde los subíndices (*i,j*) representan la posición de cada neurona y (*k,l*) la de sus neuronas vecinas, W_{ijkl} es una función asociada a la conexión entre neuronas, *f* el factor de decaimiento de la señal *Feeding*, *g* la constante de decaimiento para el umbral dinámico y *h* un potenciador de activación. Pese a esta modificación, en la práctica los parámetros involucrados en este modelo computacional continúan limitando su rendimiento ocasionando que surjan distintas estrategias de optimización [16-18], es ahí donde surge el término "conexiones nulas" [19] que más allá de realizar una elección automática de los parámetros de ajuste lo que se busca es reducir al mínimo el número de variables eliminando la influencia de los vecinos de cada neurona, simplificando aún más el modelo original de una red pulso acoplada a través de las ecuaciones (4)-(6):

$$\mathrm{F(n)} = \mathrm{f\ F(n-1)} + \mathrm{S}, \tag{4}$$

$$\mathrm{E(n)} = \mathrm{g\ E\ (n-1)} + \mathrm{h\ Y\ (n-1)}, \tag{5}$$

$$\mathrm{Y(n)} = \begin{cases} 1, & \mathrm{F(n)>E(n)}, \\ 0, & \mathrm{en\ caso\ contrario.} \end{cases} \tag{6}$$

Entre los aportes más destacados se encuentras los de Jin, X. et al. [20], quienes demostraron que la información de este tipo de red emite puede ser analizada por la condición de activación continua de cada neurona, lo que permite restringir los parámetros al satisfacer la ecuación (7):

$$S > h\frac{1-f}{1-g}, \tag{7}$$

donde S representa la imagen de entrada y *f*, *g* y *h* los valores de ajuste.

2.2. Perceptrón multicapa

Un perceptrón multicapa (MLP, *Multilayer perceptron*) es un tipo de red neuronal artificial que se caracteriza por tener en su configuración al menos una capa de entrada, una capa oculta y una capa de salida conformada por dos o más neuronas conectadas entre sí.

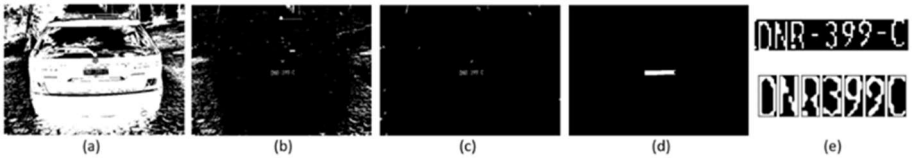


Fig. 1. Proceso de filtrado de objetos en una (a) imagen binaria (b) por área, (c) dimensiones y relación de aspecto, (d) forma geométrica y longitud y (e) segmentación utilizando la red ICMNI para definir los caracteres.

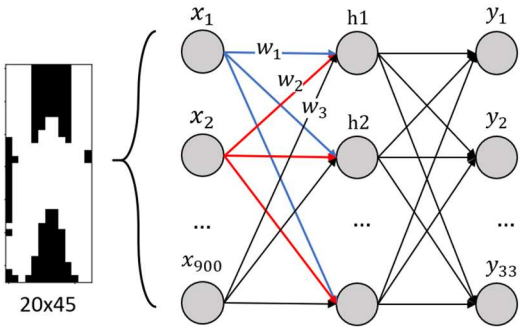


Fig. 2. Arquitectura inicial del perceptrón multicapa.

Durante el proceso conocido como *feedforward* o "propagación hacia adelante" cada neurona recibe como entrada las salidas de todas las neuronas de la capa anterior para poder obtener una salida (en este proceso se realiza una sumatoria ponderada de todos los datos y se aplica una función de activación generalmente del tipo sigmoide [21]) que dependiendo del tipo de análisis que se realice, se podrá interpretar para resolver problemas de regresión o de clasificación binaria/multiclase.

El entrenamiento se puede llevar a cabo mediante la retropropagación del error (*backpropagation*) entre capas a través del descenso del gradiente [22] para que los pesos se actualicen en función del aporte en que han contribuido a ese error y como función de coste global se suele utilizar el error cuadrático medio, que representa la suma de los errores resultantes de la diferencia entre la salida ideal y la salida de la red.

Generalmente no se alcanza un error absolutamente igual a cero, pero se puede definir un límite de error cercano a ese valor o un número de iteraciones máximas también conocidas como épocas.

3. Método propuesto

3.1. Localización y segmentación de los caracteres

El algoritmo que realiza el proceso de localización está basado en la red ICMNI para binarizar la imagen de entrada de acuerdo con lo siguiente:

1. Estandarizar las dimensiones de todas las imágenes (S) y convertirlas a escala de grises para su posterior manipulación.

Fig. 3. Ejemplos del conjunto de entrenamiento.

2. Normalizar entre 0 y 1 los niveles de intensidad de S. Esto se logra dividiendo el valor de cada píxel entre el máximo nivel de intensidad dentro de la imagen.
3. Inicializar los valores de Y, F y E en cero con las dimensiones de S.
4. Definir los parámetros de la red condicionados por la ecuación (7) y sumar la constante obtenida a la imagen normalizada.
5. De acuerdo con el comportamiento de la red establecer un número máximo de iteraciones.
6. Almacenar los pulsos de salida para analizarlos y verificar que todas las neuronas se activen al menos una vez (estadística de disparo).
7. Al término de las N iteraciones aplicar el operador unión a los pulsos obtenidos para generar una sola imagen de salida.

Extracción de las regiones de interés. Para ubicar los caracteres dentro de la imagen se realizó una serie de filtrado de objetos (véase la Fig.1) utilizando como referencia características geométricas [23]. Primero eliminando el área más grande correspondiente al vehículo empleando la técnica de componentes conectados y después limitando las dimensiones y la relación de aspecto de los posibles candidatos con *bounding boxing*.

Posteriormente a los objetos resultantes se les aplicó el operador dilatación para extraer los caracteres que pudieran haber sido descartados en la etapa anterior utilizando como estructura de referencia una matriz de 1x25.

Tras obtener la región de la Fig.1(d), se filtraron nuevamente los objetos de la zona tomando como referencia la imagen binaria original para eliminar los guiones entre caracteres y por fin aislar cada dígito y letra utilizando una vez más el operador dilatación con una nueva estructura de referencia de 50x1. Por último, se empleó la red ICMNI en los caracteres por separado para facilitar el proceso de reconocimiento.

3.2. Reconocimiento de la matrícula

Para la elección de la arquitectura del MLP (Fig.2) se realizaron diversas pruebas del algoritmo desarrollado en el lenguaje de programación de Python considerando una sola capa oculta, 900 neuronas de entrada correspondientes al total de píxeles de la imagen binaria y 33 neuronas de salida que representan el número de clases posibles.

Para ingresar las muestras a la red, primero se categorizó cada imagen en carpetas según el número o letra que representaba (0,1,2,3, etc.), después se convirtieron a escala de grises normalizando los valores entre 0 y 1 y, por último, se manipularon como vector para añadirlos a una matriz que definiría el conjunto de entrenamiento y validación. El entrenamiento se llevó a cabo por los siguientes pasos:

1. Revolver aleatoriamente el conjunto de muestras y destinar 1/5 parte al set de validación.
2. Asociar todos los vectores de entrada con su salida deseada $\{x_i, y_i\}$ utilizando la codificación *one hot.*

3. Asignar valores aleatorios pequeños a los pesos {w}.
4. Definir una tasa de aprendizaje {η} y un momento {α}.
5. Para cada muestra de entrenamiento obtener la salida de la red:

$$ai = \sum_{i=1}^{n} w_n x_n \; ; \; y_{real} = h(ai). \tag{8}$$

6. Evaluar el error cuadrático medio:

$$ECM = \sum \frac{1}{2} * (y_{ideal} - y_{real})^{2}. \tag{9}$$

7. Calcular el incremento parcial de los pesos utilizando el descenso del gradiente y la regla de la cadena:

$$\nabla E = \frac{\partial E}{\partial wo} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial ao} \cdot \frac{\partial ao}{\partial wo}, \tag{10}$$

$$\frac{\partial E}{\partial wh} = \frac{\partial E}{\partial h} \cdot \frac{\partial h}{\partial ai} \cdot \frac{\partial ai}{\partial wh}. \tag{11}$$

8. Actualizar los pesos de la capa de salida:

$$wo = wh + \alpha(wo - wo) - \eta \cdot \frac{\partial E}{\partial wo}. \tag{12}$$

Actualizar los pesos de la capa oculta:

$$wh = wh + \alpha(wh - wh) - \eta \cdot \frac{\partial E}{\partial wh}. \tag{13}$$

9. Obtener el mínimo coste global y definir un número máximo de épocas.

Al finalizar este proceso, para asignar etiquetas a nuevas muestras solo resta obtener la salida de la red con la ecuación (8) utilizando los pesos actualizados.

4. Resultados

4.1. Dataset

En virtud de la inexistencia de una base de datos con características mexicanas disponible al público, se creó un dataset a partir de imágenes frontales y traseras de automóviles capturadas en video a través de un teléfono celular con una resolución de 13 megapíxeles bajo diversas condiciones de iluminación por la hora del día a la que fueron tomadas.

En total, se logró reunir 730 imágenes las cuales solo se utilizaron para segmentar los caracteres y añadirlos al conjunto de entrenamiento del MLP. En la Fig.3 se muestran algunos ejemplos de los caracteres utilizados para el entrenamiento. El dataset

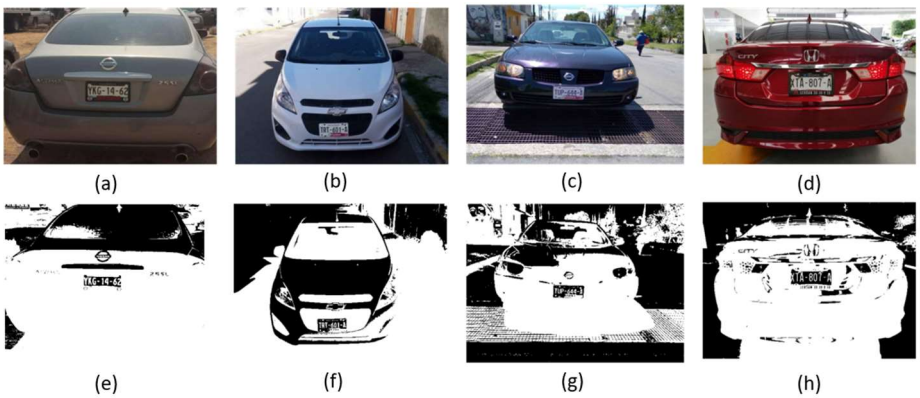


Fig. 4. Resultados del proceso de segmentación utilizando el método propuesto.

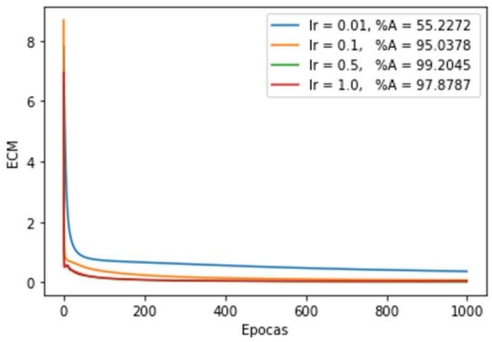


Fig. 5. Evaluación del error de aprendizaje en función de las épocas.

incluye los dígitos del 0 al 9 y las letras mayúsculas del alfabeto latino de la A a la Z excluyendo la I, Ñ, O y la Q.

Para evaluar la metodología propuesta, se creó otro dataset con características similares al primero para evitar mostrarle al algoritmo ejemplos que sirvieron para su aprendizaje. Esta nueva base de datos está conformada por 660 imágenes provenientes de diversos sitios web y al igual que el anterior el formato es compatible al asignado al estado de Puebla.

Aumento de datos. Debido al balanceo de las clases, del conjunto de datos destinado solo se pudieron seleccionar 100 ejemplos por etiqueta, lo cual afectó el entrenamiento del MLP disminuyendo el porcentaje de aciertos. Por esta razón se resolvió mejorar la robustez del algoritmo empleando la técnica de dilatación utilizando la estructura de referencia de la ecuación (14) sobre los 3,300 ejemplos para duplicar la cantidad de muestras:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \tag{14}$$

Tabla 1. Resumen de las pruebas de entrenamiento al cambiar los parámetros.

Tasa de aprendizaje	Neuronas ocultas	Momento	Épocas	Predicción correcta (%)		ECM	Tiempo
				Entrenamiento	Validación		
22 imágenes x 33 clases, total: 726, validación: 145							
0,3	50	0,5	10,000	100,0	77,931	0,0011	3 min
0,3	50	0,5	30000	100,0	83,448	0,0000	8 min
0,3	50	0,5	50,000	100,0	79,310	0,0000	16 min
100 imágenes x 33 clases, total: 3300, validación: 660							
0,1	180	0,2	30,000	99,8106	95,4545	0.0092	57 min
200 imágenes x 33 clases, total: 6600, validación: 1320							
0,5	250	0,5	30,000	99,981	98,0303	0.0013	2h

Resultados de la localización y segmentación

A través de una serie de pruebas se lograron determinar los parámetros que satisfacen la condición de activación continua (ecuación 15), siendo f = 0.85, g = 0.5 y h = 15 los valores que a su vez permiten visualizar en algún pulso la matrícula durante el proceso de segmentación:

$$S > h\frac{1-f}{1-g} = 15\,\frac{0.15}{0.5} = 4.5. \tag{15}$$

Sumar 4.5 a la imagen normalizada además de activar todas las neuronas secuencialmente, también permite determinar el número máximo de iteraciones, dado que a partir de cierta iteración la red comienza a repetir los patrones de salida.

Como se observa en la Fig.4, la implementación de la metodología muestra ser efectiva en imágenes que son variantes entre sí respecto a los niveles de intensidad que la conforman, ya sea por la textura o el ruido implícito. Durante el entrenamiento del MLP se probaron diferentes parámetros respetando la arquitectura que ya había sido determinada en la sección 3.2.

De la Tabla 1 se deduce que el aumento de épocas aumenta considerablemente el porcentaje de aciertos y el número de neuronas está ligado a la cantidad de datos que estén disponibles. El tiempo reportado corresponde a la ejecución del script implementado en el lenguaje de Python utilizando el entorno de desarrollo de Spyder con un procesador Intel i7 y 8GB de memoria RAM. La tasa de aprendizaje se fijó considerando el comportamiento de la Fig.5 evaluando el decremento del error cuadrático medio y el aumento del porcentaje de aciertos para descartar un subajuste o sobreentrenamiento del MLP.

La métrica utilizada para evaluar la metodología en el conjunto de 660 imágenes se obtuvo dividiendo el número de matrículas que se clasificaron correctamente por el total de muestras de prueba multiplicado por 100, dando como resultado un 96.2% de precisión. Solo se tomaron como válidas las placas donde todos los caracteres habían

Tabla 2. Comparación entre técnicas existentes de reconocimiento de placas y el propuesto.

Técnicas principales y referencias	YOLOv2 + SVM [5]	RPN + R-CNN [6]	R-CNN + CTC[7]	K-means + CNN [10]	ICMNI +MLP
Año	2018	2018	2020	2020	2021
Entrenamiento	8,776	10,873	2,700k	43,615	6,600
Validación	5,850	472	300	16,185	660
Variación en distancia	✓	x	✓	x	✓
Perspectiva frontal	✓	x	x	✓	✓
Loc. placa (%)	94.23	89.12%	93.7	94.8	100
Rec. caracteres (%)	99.2	99.1	98.9	98.1	96.2
Plataforma & procesador	NVIDIA Ge Force 12GB DDR5X DRAM	--	Intel Xeon E5-2680v4 NVIDIA Tesla k80	PC i5 16 GBRAM Python, Tensorflow PyTesseract	PC i7, 8GB RAM, Python, Spyder

sido correctamente etiquetados. En este punto cabe denotar que la red ICMNI facilitó en gran medida la localización de los caracteres alfanuméricos, es por ello que se logró alcanzar un 100% en la etapa de segmentación.

La Tabla 2 muestra una comparación de rendimiento entre métodos de reconocimiento de sistemas vehiculares reportados por diferentes autores y el que se propone en este artículo, además de incluir las características de los datasets utilizados. Como se observa, las técnicas de aprendizaje profundo necesitan grandes cantidades de datos durante el entrenamiento para alcanzar un elevado porcentaje de precisión al clasificar los caracteres, siendo altamente recomendables por ser relativamente insensibles a valores con ruido.

Sin embargo, aún existe apertura a otro tipo de técnicas dado que en la mayoría de los casos se consideran dependientes del hardware. El algoritmo propuesto logra estar al nivel de una red convolucional en imágenes que presentan variación en distancia sin incidir en placas sumamente borrosas y considerando únicamente una perspectiva frontal, por lo que puede utilizarse en aplicaciones que no son tan demandantes como lo son las identificaciones vehiculares múltiples [7].

En comparación a los otros autores, se logró alcanzar un 100% en la localización de la placa al aprovechar que la red ICMNI activa las neuronas según el umbral óptimo durante el proceso de binarización. Esta propiedad permitió obtener una segmentación más robusta en todo el dataset utilizando los mismos parámetros para todas las imágenes sin necesidad de agregar algún filtro (lineal o no lineal) para reducir el impacto del ruido o un espacio de color para aumentar el contraste, procesos que forman parte del preprocesamiento que se suele añadir en la etapa de localización de la placa.

También se alcanzó un porcentaje de reconocimiento de caracteres cercano a las otras contribuciones, por lo que se infiere que este valor puede mejorar aumentando el número de ejemplos para entrenar la red neuronal.

5. Conclusiones

De acuerdo con los resultados obtenidos, la metodología propuesta presenta un buen desempeño en imágenes donde los niveles de intensidad no son homogéneos, siendo útil en casos donde la localización de los caracteres se ve afectado por cierto tipo de

sombras o por cambios bruscos de iluminación, sin embargo, es susceptible a la distancia de captura y la inclinación de la placa.

Este enfoque basado en la localización de la matrícula y no de la placa sirve como alternativa en casos donde no existe un color primario predominante en la zona de la placa que permita aislarla con facilidad o cuando el valor del umbral es variante en todas las imágenes. El proceso de reconocimiento no se ve limitado a un número determinado de caracteres por lo que podría utilizarse en imágenes que incluyan otro tipo de vehículos, como autobuses o motocicletas fijando un rango de distancia de captura y un grado de inclinación adecuado para este sistema.

Para trabajos futuros se considera ajustar la matrícula a un solo perfil para incluir en el dataset de prueba imágenes rotadas o con perspectiva lateral para aumentar el desempeño del algoritmo en otro tipo de escenarios, reduciendo a su vez las restricciones.

Referencias

1. Yimyam, W., Ketcham, M.: The automated parking fee calculation using license plate recognition system. In: Proceedings of International Conference on Digital Arts, Media and Technology (ICDAMT), IEEE pp. 325–329 (2017)
2. Zhu, L., Wang, S., Li, CH., Yanget, Z.: License plate recognition in urban road based on vehicle tracking and result integration. Journal of Intelligent Systems. vol. 29, no. 1, pp. 1587–1597 (2020)
3. Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C.: Automated license plate recognition: A survey on methods and techniques. IEEE Access, vol. 9, pp. 11203–11225 (2020)
4. Weihong, W., Jiaoyang, T.: Research on license plate recognition algorithms based on deep learning in complex environment. IEEE Access, vol. 8, pp. 91661–91675 (2020)
5. Lin, C. H., Lin Y. S., Liu, W. C.: An efficient license plate recognition system using convolution neural networks. In: Proceedings of IEEE International Conference on Applied System Invention (ICASI), pp. 224–227 (2018) doi: 10.11 09/ICASI.2018.8394573
6. Yang, Z., Du, F. L., Xia, Y., Zheng, C. H., Zhang, J.: Automatic license plate recognition based on faster R-CNN algorithm. In: Proceedings of International Conference on Intelligent Computing, Springer, Cham, pp. 319–326 (2018) doi: 10.1007/978-3-319-95957-3_35
7. He, M. X., Hao, P.: Robust automatic recognition of Chinese license plates in natural scenes. IEEE Access, vol. 8., pp. 173804–173814 (2020)
8. Anagnostopoulos, C. N. E., Anagnostopoulos, E. I., Loumos, V., Kayafas. E.: A license plate-recognition algorithm for intelligent transportation system applications. In: Proceeding of IEEE Transactions on Intelligent transportation systems. vol. 7, no. 3, pp. 377–392 (2006) doi: 10.1109/TITS.2006.880641
9. Jalil, N. A., Basari, A. S. H., Salam, S., Ibrahim, N. K., Norasikin, M. A.: The utilization of template matching method for license plate recognition: A case study in Malaysia. In: Proceedings of Advanced Computer and Communication Engineering Technology. Springer, Cham, pp. 1081–1090 (2015) doi: 10.1007/ 978-3-319-07674-4_100
10. Pustokhina, I. V., Pustokhin, D. A., Rodrigues, J. J. P. C., Gupta, D., Khanna, A., Seo, C., Joshi G. P.: Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems. IEEE Access, vol. 8, pp. 92907-92917 (2020)
11. Escalante, S., Murray, V.: Automatic recognition of peruvian car license plates. In: IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), IEEE, pp. 1–4 (2020)

12. Ma, Y.: Applications of pulse-coupled neural networks. Higher Education Press, Beijing, China (2010)
13. Lindblad, T., Kinser, J. M.: Image processing using pulse-coupled neural networks. Heidelberg: Springer (2005)
14. Mohsen, H., El-Dahshan, E. S. A., Salem, A. B. M.: A machine learning technique for MRI brain images. In: Proceedings of 8th International Conference on Informatics and Systems (INFOS), IEEE, pp. BIO-161–BIO-165 (2012)
15. Mokhayeri, F., Akbarzadeh-T, M. R.: A novel facial feature extraction method based on ICM network for affective recognition. In: Proceedings of International Joint Conference on Neural Networks, IEEE, pp. 1988–1993 (2011)
16. Dharwal, D., Shanker, R., Bhattacharya, M.: Automatic parameter setting of pulse coupled neural network for image segmentation. In: International Conference on Communication and Signal Processing (ICCSP), IEEE, pp. 2157–2161 (2016)
17. Hernández, J., Gómez, W.: Automatic tuning of the pulse-coupled neural network using differential evolution for image segmentation. In: Mexican Conference on Pattern Recognition, Springer, Cham, pp. 157–166 (2016)
18. Deng, X., Ma, Y.: PCNN model analysis and its automatic parameters determination in image segmentation and edge detection. Chinese Journal of Electronics, vol. 23, no. 1, pp. 97–103 (2014)
19. Prieto, C., Rodríguez, A.: A soft image edge detection approach based on the time matrix of a PCNN. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, pp. 463-469 (2008)
20. Jin, X., Zhou, D., Jiang, Q., Chu, X., Yao, S., Li, K., Zhou, W.: How to analyze the neurodynamic characteristics of pulse-coupled neural networks? A Theoretical Analysis and Case Study of Intersecting Cortical Model, IEEE, Transactions on Cybernetics, vol. 52, no. 7, pp. 6354–6368 (2021)
21. Nunes, I., Hernane, D.: Artificial neural networks: A practical course. Springer (2018)
22. Isasi, P., Galván, I.: Redes de neuronas artificiales. Un enfoque práctico, Pearson Educación, Madrid, España (2004)
23. Cuevas, E., Zaldivar, D., Perez-Cisneros, M.: Procesamiento digital de imágenes con Matlab y Simulink. Alfaomega, vol. 10, no. 1, pp. 77–78 (2010)

ISSN 1870-4069

Sistema de monitoreo remoto de temperatura y humedad utilizando dispositivos móviles e IOT

José Ramon Arratia-Zapata, Marco Aurelio Nuño-Maganda,
Yahir Hernández-Mier, Said Polanco-Martagón

Universidad Politécnica de Victoria,
Laboratorio de Sistemas Inteligentes,
México

{mnunom, yhernandezm, spolancom}@upv.edu.mx

Resumen. En el presente trabajo se presenta un sistema para el monitoreo de temperatura y humedad, basado en un microcontrolador con un stack WiFi, que envía la información de sensores a un MQTT. Mediante una aplicación móvil desarrollada nativamente para Android, se realiza la lectura de la información alojada en la nube mediante un protocolo MQTT y se despliegan gráficas de monitoreo en tiempo-real, así como el historial de mediciones. Se probó el esquema propuesto en un ambiente de vivero simulado y los resultados muestran que la aplicación nativa desarrollada permite la lectura y visualización de los datos sin problemas. Esta aplicación podría ser de utilidad en todos los procesos que requieran el monitoreo continuo de temperatura y humedad, como en procesos de almacenamiento de productos agrícolas, comida y productos farmacéuticos.

Palabras Clave: Aplicación móvil, MQTT, internet de las cosas, microcontrolador, DHT22.

Remote Temperature and Humidity Monitoring System Using Mobile Devices and IOT

Abstract. In the present work, a system for monitoring temperature and humidity is presented, based on a microcontroller with a WiFi stack, which sends information from sensors to an MQTT. Through a mobile application developed natively for Android, the information stored in the cloud is read using an MQTT protocol and real-time monitoring graphs are displayed, as well as the measurement history. The proposed scheme was tested in a simulated nursery environment and the results show that the developed native application allows the reading and visualization of the data without problems. This application could be useful in all processes that require continuous monitoring of temperature and humidity, such as storage processes for agricultural products, food and pharmaceuticals.

Keywords: Mobile application, MQTT, internet of things, microcontroller, DHT22.

1. Introducción

Los avances tecnológicos en el campo del internet de las cosas (IoT, por sus siglas en inglés) hacen posible controlar y monitorear en cualquier parte del mundo, ambientes y equipos. Actualmente, existen varias alternativas de domótica para realizar dicha actividad que permiten reducir la intervención del usuario incluso reducirla casi por completo. Estos avances se han acelerado en la última década y sus costos se han reducido tanto, que hoy en día es posible acceder a esta tecnología con muy poco dinero.

El IoT es un concepto que existe desde 1999 y fue propuesto por Kevin Ashton. Desde la expansión masiva del Internet en 1995, hasta el día de hoy, las conexiones no han dejado de incrementarse y globalizarse.

Al principio solo se podía acceder a Internet desde una computadora, pero hoy en día, gracias al aumento de ancho de banda de la red y la accesibilidad de nuevas tecnologías de computación más baratas y eficientes, es posible que dispositivos no computarizados o que simplemente que no fueron diseñados para que se puedan conectar a otros dispositivos, como los electrodomésticos, ahora pueden ser utilizados y monitoreados en forma remota. A esto se le conoce como IoT [1, 2].

El monitoreo remoto de temperatura y humedad es importante en diversas áreas, como la agrícola [3, 4], la farmacéutica [5, 6] o la alimentaria [7, 8]. Con la tecnología actual es posible realizar este monitoreo con diversos dispositivos y diversas tecnologías de recolección y visualización de información.

Un trabajo relacionado con el sistema propuesto en este artículo es el de [9], donde los autores proponen un sistema basado en IoT para el monitoreo de temperatura usando un sensor no invasivo. Proponen también el desarrollo de una aplicación móvil usando un protocolo de envío de datos y una base de datos no especificada.

Otro trabajo relacionado se puede encontrar en [10], donde los autores desarrollan una aplicación móvil basada en hardware y software abiertos para el monitoreo de la cadena de frío en el transporte y almacenaje de comida. En este trabajo, al igual que en el propuesto, se utiliza la plataforma en la nube de ThingSpeak para el almacenaje y visualización de la información enviada por los sensores.

En este trabajo se propone una alternativa al uso de plataformas para la visualización de información adquirida en sistemas de IoT en la forma de una aplicación nativa desarrollada en el sistema operativo Android, la cual permite visualizar la información adquirida por sensores de temperatura y humedad, en conjunto con un microcontrolador con un stack WiFi integrado, que envían información a una plataforma de almacenaje y gestión de datos en la nube.

Esta aplicación forma parte de un sistema completo de IoT para el monitoreo de la humedad y temperatura, el cual puede ser aplicado a refrigeradores, cuartos fríos, viveros, y cualquier área donde sea necesario vigilar estos dos parámetros.

Por ejemplo, en el sector de salud se puede aprovechar esta tecnología para mantener los medicamentos con una humedad y temperatura adecuadas para su conservación, evitando mermas en medicamentos. En la agricultura, se podría usar para controlar de forma continua la humedad y la temperatura en silos de almacenamiento.

El sistema propuesto está basado en un microcontrolador Arduino, el cual, con ayuda de un chip ESP8266, envía los datos adquiridos por un sensor de temperatura y humedad a un *broker*, disponible gratuitamente en internet, en donde pueden ser leídos

por dispositivos Android utilizando el protocolo Message Queuing Telemetry Transport (MQTT), y presentados en forma de gráficas a los usuarios.

2. Materiales y métodos

2.1. Interfaz de programación de aplicaciones (API)

En la actualidad, cuando se desarrollan aplicaciones, sin importar el contexto, se hace uso de APIs (Interfaces de Programación de Aplicaciones, por sus siglas en inglés), las cuales facilitan el proceso de diseño y desarrollo de la aplicación móvil. Una API es un software que permite que las aplicaciones se comuniquen entre sí.

Es un conjunto de funciones que permiten a las aplicaciones acceder a datos e interactuar con componentes de software externos, sistemas operativos o microservicios. Una API pasa una solicitud de un usuario a un sistema y le devuelve la respuesta del sistema. Esto permite a los desarrolladores extraer datos de diferentes formas y para diversos propósitos, dependiendo de la funcionalidad que se requiera [11].

2.2. Protocolo MQTT

Es un protocolo de transporte de mensajes entre el cliente y un servidor el cual está basado en publicaciones y suscripciones, es creado por IBM con el propósito de recolectar información, es un protocolo utilizado para el intercambio de mensajes entre dispositivos que trabajan con IoT, permite la conectividad Máquina a Máquina (Machine to Machine por sus siglas en inglés M2M) e IoT. Este protocolo de mensajería ligera la cual es esquematizado en base a un servidor broker con la funcionalidad de publicación – suscripción que se emplea sobre el protocolo TCP/IP [12, 13].

2.3. Servidor MQTT (Bróker)

Un servidor MQTT se encarga de administrar el protocolo, el flujo de los mensajes de cada nodo de un sensor, además del estado del servicio. En el mercado existen distintas formas de software y proveedores que implementan el servidor MQTT, como por ejemplo Google Cloud, Thingspeak o Mosquitto. En el sistema propuesto se usó ThingSpeak como el servidor MQTT, debido a que la interfaz de lectura y escritura de datos a la plataforma es amigable. ThingSpeak tiene soporte de la empresa Mathworks. ThingSpeak es un aplicación y API de IoT que permite almacenar y obtener datos de IoT mediante protocolos de internet como el HTTP y MQTT, a través de peticiones web.

2.4. Sensores de temperatura/humedad empleados

En este trabajo se emplearon dos diferentes sensores de temperatura/humedad. El primero fue el DHT-11, el cual integra un sensor capacitivo que mide la humedad, y un termistor para medir la temperatura. El segundo fue el sensor DHT-22, cuyo funcionamiento es similar al DHT-11, pero tiene mayor resolución y puede ser

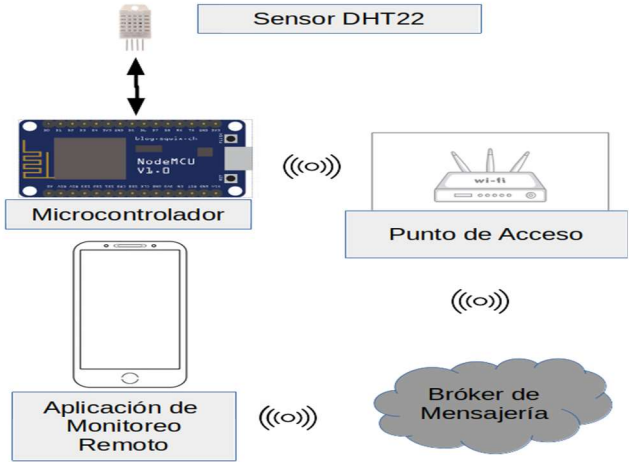


Fig. 1. Diagrama de bloques de los componentes del sistema.

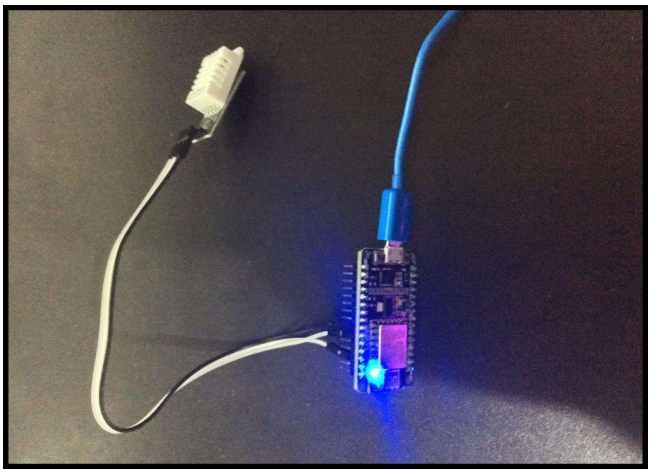


Fig. 2. Sensor DHT22 conectado al NodeCMU.

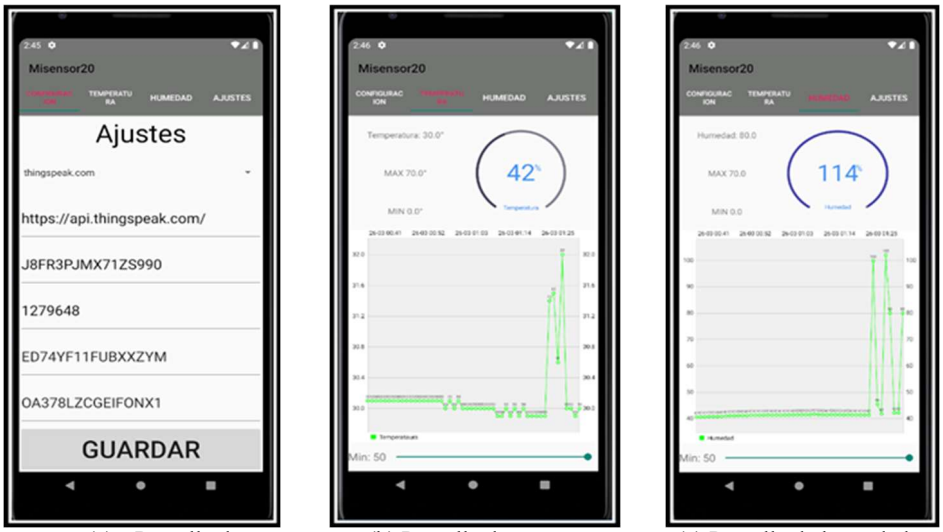
alimentado a 3.3V. El DHT-11 tiene una exactitud de ±2ºC, mientras que el DHT-22 cuenta con una exactitud de ±0.5ºC.

El sistema propuesto involucra los siguientes componentes de hardware:

- Una fuente de alimentación,
- Un sensor DHT22,
- Una tarjeta microcontroladora NodeMCU,
- Un punto de acceso a internet. Este es un requerimiento obligatorio del proyecto, ya que los datos son enviados vía WiFi.
- Un teléfono inteligente.

Tabla 1. Resultados de sensado de temperatura y humedad.

	Temperatura	**Humedad**
Habitación Climatizada	28.4	42.1
	28.4	44.1
	28.5	43.1
	28.6	42.9
	28.6	43.9
Maceta de Jardín	30.6	76.1
	31.6	53.0
	31.5	54.3
	31.6	53.4
	31.7	54.1



(a) Pantalla de configuración (b) Pantalla de temperatura (c) Pantalla de humedad

Fig. 3. Pantallas de a la aplicación de monitoreo.

Para el desarrollo de la interfaz nativa en Android se usaron las siguientes herramientas de software:

- Android Studio.
- Arduino IDE.
- Biblioteca DHT.
- Biblioteca MPAndroidChart. Esta biblioteca permite mostrar de forma fácil y personalizable grandes conjuntos de datos en forma de gráficas [14].

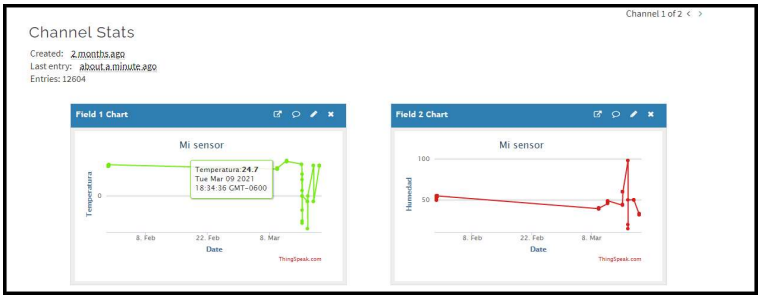
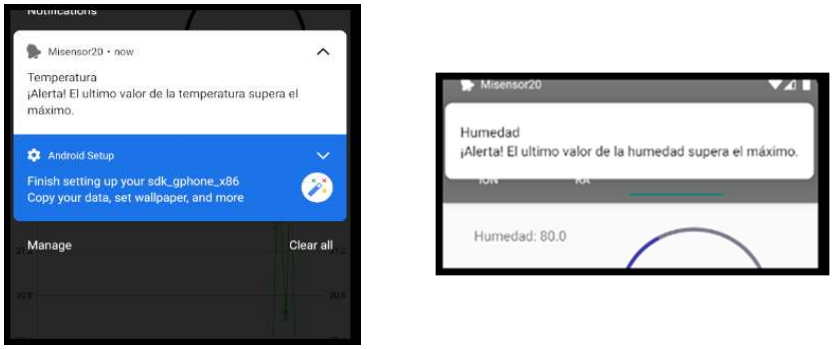


Fig. 4. Gráficas de temperatura y humedad a partir de los datos enviados por el sensor hacia el servidor.



(a) Alerta de Temperatura (b) Alerta de Humedad

Fig. 5. Ejemplos de alertas recibidas mediante la aplicación desarrollada.

Fig. 6. Monitorización de la humedad de una planta.

- Biblioteca CircleProgress. Esta biblioteca permite la creación de gráficas de pastel para representar un porcentaje [15].
- Retrofit. Es un cliente REST para aplicaciones Android.
- Broker de mensajería. Para el desarrollo del proyecto se evaluaron los siguientes brokers: Google Cloud, Microsoft Azure IoT Suite, IBM Watson IoT Platform, AWS IoT Platform y Thingspeak. Después de evaluar estos brokers, se optó por utilizar Thingspeak, debido a que la interfaz de lectura y escritura de datos a la plataforma en la nube es de fácil uso y el límite de datos en la versión de prueba es suficiente para las pruebas realizadas.

3. Sistema propuesto

El sistema de monitoreo propuesto obtiene las lecturas de los sensores en tiempo real, y que estos datos sean visibles por un usuario final mediante una aplicación móvil. Para lograr este objetivo, un microcontrolador lee continuamente datos de los sensores y los envía continuamente a través de internet. El broker de mensajería concentra los datos de los sensores, y continuamente envía los datos hacia la aplicación móvil suscrita a los canales disponibles en el broker.

El usuario configura la aplicación móvil para establecer los umbrales de las lecturas que para el dominio particular representan valores fuera de rango. Una vez configurada la alerta, al registrarse una lectura fuera del rango, una notificación es desplegada en todos los dispositivos donde la aplicación haya sido instalada, independientemente si la aplicación está abierta o no. La aplicación permite al usuario desplegar gráficas del comportamiento de la temperatura y humedad a lo largo del tiempo. En la figura 1, se muestran los componentes del sistema previamente descritos.

3.1. Sensores de temperatura/humedad empleados

El sensor DHT22, es un sensor que requiere de una alimentación de 3.3 a 6 Volts, además de una salida de tierra. Este sensor tiene un precio bastante accesible además de ser muy eficiente con respecto a la lectura de datos y con un rango mayor de precisión que su antecesor (el DHT11).

Por otro lado, el NodeMCU está integrado por varios pines de salida de 3.3 Volts lo suficiente para mantener un buen funcionamiento en relación con el sensor y varios pines de tierra GDN, de esta forma el microcontrolador puede realizar la lectura de temperatura y humedad, sin embargo, para llevar a cabo dicha tarea es necesario conectar el sensor a uno de los pines de entrada y salida del NodeMCU. La conexión correcta entre el sensor y el NodeCMU se muestra en la figura 2.

3.2. Diseño y desarrollo de aplicación móvil de monitoreo remoto

La aplicación propuesta fue diseñada para incluir múltiples pestañas para una mejor organización. Esta aplicación emplea el protocolo MQTT (Message Queue Telemetry Transport), el cual es un protocolo diseñado específicamente para las aplicaciones IoT como la descrita en este artículo.

El protocolo MQTT funciona mediante publicaciones y suscripciones a un tema (modelo), mediante un Broker que gestiona las publicaciones y suscripciones. En la primera pestaña (mostrada en la figura 3(a)), el usuario configura los datos de la conexión al servidor para la adquisición de los datos. En esta pestaña es posible configurar la cantidad de datos que serán visualizados.

En las figuras 3(b) y 3(c), se muestran diferentes pestañas de la aplicación que incluyen la última medición y una gráfica de líneas mostrando el histórico de los datos de temperatura y humedad. Se muestran dos gráficas diferentes en cada una de las pestañas mostradas.

– La gráfica circular es utilizada para representar de forma coloreada el porcentaje entre un rango mínimo y máximo usados para el control de la temperatura o la

humedad. Para obtener esta gráfica se realiza una simple operación matemática la cual nos dará como resultado el porcentaje el cual se le enviará a un objeto de la clase ArcProgress, al igual que la gráfica de líneas debe pasar por todos los procesos desde la creación en la vista de la aplicación hasta ser referenciado, la única diferencia es que este solo requiere un valor entero el cual será el porcentaje a colorear y se podrá visualizar en su centro.

– La gráfica de líneas es desplegada usando el conjunto de datos obtenido por retrofit y la librería MQTT, y vaciando los datos hacia un objeto Dataset.

4. Resultados

Con respecto al Bróker de mensajería, se configuró la conexión del microcontrolador con el servidor, de tal forma que fue posible la visualización en el servidor de las lecturas reportadas por el sensor.

Se procedió a verificar que los datos sean visibles en el servidor y en la aplicación desarrollada. Para esto, se utilizaron las gráficas predefinidas para el muestreo de los datos en el sitio Thingspeak para el proyecto creado para la aplicación, como las mostradas en la figura 4.

La aplicación desarrollada fue probada en varios dispositivos con diferentes versiones del sistema operativo Android. La versión más antigua en la que el sistema fue probado es la versión 5.0, también se probó en las versiones 6.0, 7.0, 8.0 y la versión más reciente disponible al momento de probar el sistema (Android 11.0).

Para efectos de envío de notificaciones, el usuario puede configurar el lapso de tiempo y la frecuencia para el envío de una o varias alertas estas se manifiestan en forma de mensaje y una alerta de sonido y vibración del dispositivo, los mensajes se mostrarán en el apartado de notificaciones del dispositivo.

En la figura 5 se muestran dos ejemplos de estas notificaciones en dispositivos diferentes, donde para la temperatura se muestra la figura 5(a) y para la humedad se muestra la figura 5(b).

Finalmente, se hicieron mediciones en dos diferentes escenarios: una habitación climatizada y una maceta de jardín. Las mediciones obtenidas se muestran en la tabla 1. Estas pruebas permitieron simular las condiciones necesarias para un par de escenarios más realistas: monitorear de manera remota las condiciones de humedad y temperatura de un invernadero, y monitorear la temperatura de refrigeradores con aplicación en agricultura y salud.

En la figura 6, se muestra el sensor montado en una maceta, de donde se obtuvieron las lecturas reportadas en la tabla.

5. Conclusiones y trabajo futuro

Con base a las pruebas realizadas, fue posible corroborar que el sistema realiza las funciones para las cuales fue diseñado: la medición de temperatura y de humedad. Para validar el sistema propuesto, se verificaron dos posibles escenarios: el monitoreo remoto de refrigeradores y la detección de los niveles de humedad en un huerto remoto.

La aplicación móvil es capaz de mostrar información que alerta al usuario acerca de anomalías en las lecturas, incluyendo la configuración de la aplicación para desplegar al usuario notificaciones informando de estas anomalías.

Además, la aplicación despliega el histórico de las mediciones obtenidas en forma gráfica, lo cual permite al encargado del monitoreo tener un panorama más amplio de las variaciones en las mediciones.

Algunas mejoras que pueden realizarse consisten en el aumento del número de sensores, mejorar la interfaz gráfica y optimizar la conectividad y la obtención de datos de los sensores.

Referencias

1. Uribe-Castro, A.: Análisis del nivel de seguridad presente en los dispositivos que componen el internet de las cosas (2020)
2. Xia, F., Yang, L. T., Wang, L., Vinel, A.: Internet of things. International Journal of Communication Systems, vol. 25, no. 9, pp. 1101–1102 (2012)
3. Karthikeyan, P. R., Chandrasekaran, G., Kumar, N. S., Sengottaiyan, E., Mani, P., Kalavathi, D. T., Gowrishankar, V.: Iot based moisture control and temperature monitoring in smart farming. Journal of Physics: Conference Series, no. 1, pp. 1–7 (2021)
4. Mabrouki, J., Azrour, M., Dhiba, D., Farhaoui, Y., Hajjaji, S. E.: An intelligent IoT-based food quality monitoring approach using low-cost sensors. Big Data Mining and Analytics, vol. 4, no. 1, pp. 25–32 (2021)
5. Widjaya, D.: Enhancing vaccine refrigerator temperature reporting system using IoT technology. Suranaree Journal of Science and Technology, vol. 25, no. 3, pp. 225–234 (2018)
6. Roduit, B., Luyet, C. A., Hartmann, M., Folly, P., Sarbach, A., Dejeaifve, A., Dobson, R., Schroeter, N., Vorlet, O., Dabros, M., Baltensperger, R.: Continuous monitoring of shelf lives of materials by application of data loggers with implemented kinetic parameters. Molecules, vol. 24, no. 12 (2019)
7. Popa, A., Hnatiuc, M., Paun, M., Geman, O., Hemanth, D. J., Dorcea, D., Son, L. H., Ghita, S.: An intelligent IoT-based food quality monitoring approach using low-cost sensors. Symmetry, vol. 11, no. 1, pp. 18 (2019)
8. Konur, S., Lan, Y., Thakker, D., Morkyani, G., Polovina, N., Sharp, J.: Towards design and implementation of industry 4.0 for food manufacturing. Neural Computing and Applications (2021)
9. Ramalho, J. F.C. B., Carlos, L. D., André, P. S., Ferreira, R. A. S.: Moptical sensing for the internet of things: A smartphone-controlled platform for temperature monitoring. Advanced Photonics Research, vol. 2, no. 6 (2021)
10. Ramírez-Faz, J., Fernández-Ahumada, L. M., Fernández-Ahumada, E., López-Luque, R.: Monitoring of temperature in retail refrigerated cabinets applying IoT over open-source hardware and software. Sensors, vol. 20, no. 3 (2020)
11. Bierhoff, K.: Api protocol compliance in object-oriented software. PhD thesis, Carnegie Mellon University, USA AAI3370353 (2009)
12. Yassein, M. B., Shatnawi, M. Q., Aljwarneh, S., Al-Hatmi, R.: Internet of things: Survey and open issues of MQTT protocol. In: International Conference on Engineering MIS (ICEMIS), pp. 1–6 (2017)
13. Soni, D., Makwana, A.: A survey on MQTT: A protocol of internet of things (IoT). In: International Conference On Telecommunication, Power Analysis And Computing Techniques ICTPACT´17), vol. 20 (2017)
14. Jahoda, P.: MPAndroidChart. https://github.com/PhilJay/MPAndroidChart (2019)

15. LianHaiLiang, M.: Circleprogress. https://github.com/MichaelLianHaiLiang/CircleProgress (2018)

ISSN 1870-4069

Metodología para la detección de mascarilla mediante aprendizaje automático

Ma. del Carmen Santiago, Ana C. Zenteno, Yeiny Romero,
Judith Pérez, Gustavo T. Rubín, Antonio Álvarez, Alexis Meza

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computación,
México

{marycarmen.santiago, ana.zenteno, yeiny.romero, judith.perez,
gustavo.rubin}@correo.buap.mx, eduard-alvarez@live.com.mx,
alexis.meza@alumno.buap.mx

Resumen. El Covid-19 nos ha obligado a emprender acciones de protección para evitar el contagio, una de éstas es el uso de mascarillas. El reconocimiento facial (RF) tiene sus bases desde 1883, gracias a su estudio se tienen técnicas que se basan en modelos y apariencias. El RF requiere de tres etapas: a) Detección del rostro en una imagen, b) extracción de las características y c) identificación y/o verificación. En este trabajo se presenta la metodología para llevar a cabo a través del RF, la detección de la mascarilla en el rostro. Se presentan los primeros resultados favorables a partir de los cuales se proponen las bases para un proceso de optimización que busca realizar la identificación del uso correcto de la mascarilla.

Palabras clave: COVID-19, reconocimiento facial, detección mascarilla.

Methodology for Mask Detection Using Machine Learning

Abstract. Covid-19 has forced us to take protective actions to avoid contagion, one of these is the use of masks. Facial recognition (RF) has its foundations since 1883, thanks to its study there are techniques that are based on models and appearances. RF requires three stages: a) Face detection in an image, b) feature extraction, and c) identification and/or verification. This paper presents the methodology to carry out, through RF, the detection of the mask on the face. The first favorable results are presented, from which the bases for an optimization process that seeks to identify the correct use of the mask are proposed.

Keywords: COVID-19, facial recognition, mask detection.

1. Introducción

En el último año la sociedad ha sufrido un fuerte cambio en la forma de relacionarnos y de vivir nuestras vidas diarias. Esto ocasionado por la pandemia de COVID-19 donde la principal problemática ha sido la rápida propagación del virus. La situación nos obligó a aislarnos para evitar el contagio. Sin embargo, algunas personas como son los médicos corren un alto riesgo de contagio al estar expuestos al virus constantemente realizando sus trabajos, en muchos casos sin protección alguna. [1].

El reconocimiento facial es una tecnología que se encarga de identificar a un sujeto de forma no invasiva, con la ayuda de una cámara, esta tecnología se ha desarrollado con rapidez. Fue Alphonse Bertillon en 1883 quien sentó las bases del sistema de reconocimiento facial, usaba como base un sin número de medidas antropométricas como: la distancia entre los ojos, la simetría o los diferentes rasgos faciales de un individuo. Este sistema es una tendencia en el ámbito forense e incluso ha llegado al punto de tener opinión en la corte de justicia para determinar la inocencia de una persona con antecedentes penales. Su uso para reconocer personas puede agilizar el control de accesos a edificios corporativos y gubernamentales.

Esta tecnología se ha utilizado incluso para identificar criminales conocidos o sospechosos y en el ámbito socio económico, las empresas pueden analizar el rostro de los clientes para adaptar estrategias de marketing. Pero esta tecnología también enfrenta serios problemas de privacidad, ya que puede ser utilizada para rastrear a los individuos a través de sus comunidades, incluso por el mundo.

Los avances del reconocimiento facial parten del estudio de la biometría y ésta sumada a la tecnología, da como resultado la toma de medidas y el análisis de datos biológicos como el ADN, la huella dactilar, el iris y la voz. Así nacen los sistemas de reconocimiento facial, que toma sus decisiones de identificación con la ayuda de las características personales (fotografías y videos) de cada persona y plasmándolo en una imagen digital que puede ser reconocida o verificada de forma automatizada mediante un ordenador.

2. Estado del arte

En el proceso de reconocimiento facial se utilizan algoritmos para el procesamiento de imágenes donde se analizan cientos de rostros y se utiliza un mapeo facial que capta al rededor de100 expresiones faciales, todas tienen una medida de 50 x 50 píxeles de ancho y alto, por lo que el costo computacional aumenta [2].

Hay dos familias de técnicas de reconocimiento facial: técnicas basadas en la apariencia y técnicas basadas en modelos. Vea la figura 1.

2.1. Reconocimiento facial

Los sistemas basados en la apariencia se utilizan directamente sobre las imágenes, sin hacer uso de modelos 3D, cada imagen se representa como un punto en un subespacio vectorial para clasificar, para ello es necesario entrenar previamente el sistema que se utilizara con imágenes de diferentes rostros, con diferentes vistas. Por otro lado, los sistemas basados en modelos intentan construir un modelo lo más

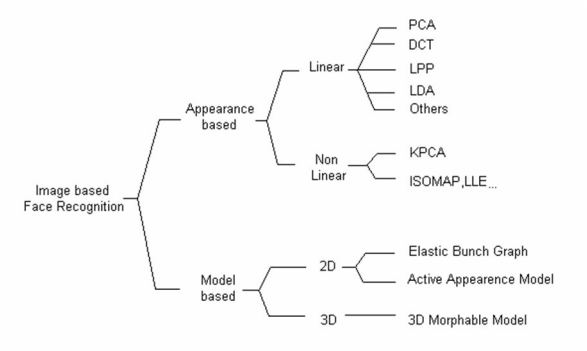


Fig. 1. Técnicas de reconocimiento facial. [3].

descriptivo posible de la cara humana, lo que permite detectar con precisión las variaciones faciales.

Estos sistemas tratan de obtener características biométricas de las imágenes para realizar el reconocimiento, el algoritmo sabe con anticipación el objeto que ha de representar y lo que intenta hacer es que corresponda el rostro real con el modelo. El proceso usado consiste en a) Construcción del modelo, b) Aplicación del modelo a la imagen de prueba y c) Uso de parámetros del modelo ajustado para calcular la similitud entre la imagen de prueba y las imágenes de referencia para realizar el reconocimiento.

Independientemente de la técnica utilizada para la solución, el RF requiere de tres etapas: a) Detección del rostro en una imagen, b) extracción de las características y c) identificación y/o verificación [5].

La detección del rostro implica encontrar las áreas dentro de una imagen que contiene un rostro. Básicamente se trata de descartar todo lo que sea fondo y así obtener la ubicación y tamaño exacto de la cara esto se puede lograr por medio de clasificación de patrones utilizando redes neuronales artificiales (perceptrón multicapa). La extracción de las características es la obtención de propiedades o parámetros particulares de cada rostro para luego poder ser clasificados.

El reconocimiento consiste en clasificar las características extraídas de cada rostro, para esto existen tres aproximaciones basadas en: a) concepto de similaridad, b) aproximación probabilista y c) optimización de criterio error.

Las aproximaciones basadas en el concepto de similaridad son las más simples e intuitivas, los patrones similares son asignados a la misma clase, se establece una métrica que define la similaridad y luego se clasifica por medio de plantillas o mínima distancia usando uno o varios prototipos de clase.

La técnica de eigenfaces aplica la regla del vecino más cercano, utilizando como métrica la distancia Euclídea. En el enfoque probabilístico, se utilizan los conceptos de la teoría de la decisión estadística para establecer los bordes de decisión de las diferentes clases. Se asume que las características, que representan a un patrón, tienen una función de densidad de probabilidad condicionada a la clase. Las reglas de decisión más conocidas: Bayes y la regla de máxima probabilidad.

La tercera aproximación se basa en construir los bordes de decisión optimizando algún criterio de error. Las redes neuronales son entrenadas con un algoritmo de entrenamiento a partir de un conjunto de datos (perceptrón multicapa). [3, 4].

2.2. Reportes de reconocimiento facial

En el Sistema de INTERPOL de Reconocimiento Facial (IFRS) se almacenan las imágenes faciales enviadas por más de 179 países, lo que la convierte en una base de datos policial de ámbito mundial única. Combinado con un software automatizado de identificación biométrica, este sistema es capaz de identificar a una persona o de comprobar su identidad mediante la comparación y el análisis de los modelos, formas y proporciones de sus rasgos y contornos faciales [6].

En la Universidad Técnica del Norte los estudiantes de la FICA (Facultad de Ingeniería en Ciencias Aplicadas) desarrollaron un prototipo que permite identificar sujetos conocidos (registrado en la base de datos) y no conocidos. Se usaron CNNs (Redes Neuronales convolucionales), y herramientas como un smartphone y una cámara de video vigilancia en tiempo real [7].

Un reporte sobre el diseño de un sistema de reconocimiento de rostros, aplicando inteligencia y visión artificial, usa un modelo para generar y establecer puntos de interés en el rostro, con el fin de entrenar una red neuronal, que permita identificar características faciales, para el reconocimiento y clasificación de sujetos. Se utilizó el algoritmo ASM (Active Shape Model – Modelo de Formas Activas) el cual genera un modelo de patrones y características que se pretenden determinar [8].

En el artículo Redes neuronales artificiales para el control de acceso basado en reconocimiento facial, se generó un prototipo de reconocimiento facial, utilizando un método basado en características geométricas. Primero obtuvieron los patrones del rostro de los usuarios para generar y construir las redes neuronales, cada determinado tiempo se obtienen variaciones de patrones del rostro de un usuario [9].

En [10] se generó un diseño de un sistema de detección de intrusos y control de acceso mediado por modelos biométricos e inteligencia artificial. Se utilizó como base el algoritmo de detección de rostros implementado por Geitgey, Deep Learning y visión computacional, apoyado sobre OpenCV. Convierte la imagen de entrada a escala de grises (0 - 255), se aplican los descriptores HOG, los cuales evalúan cada uno de los pixeles presentes en la imagen y determinan su gradiente con respecto a su entorno, para a partir de allí se realiza la cuantificación de los elementos.

En [3] se realizan pruebas con los métodos de reconocimiento facial, principalmente PCA, DCT y LPP usando bases de datos públicas para el análisis y desarrollo de las pruebas y obtención de resultados. El uso de estos métodos requiere un número de componentes para trabajar e identificar, para que estos métodos tengan el mejor rendimiento, en cada prueba se determinó que el número de componentes a utilizar según el método es: 55 para PCA, 80 para DCT y 40 para LPP, y se puede decir que el método más consistente y que ofrece un mayor índice de identificación correcta es LPP.

3. Metodología

Para llevar a cabo la detección del uso correcto del cubrebocas, se realizan las siguientes etapas en el sistema desarrollado: Detección de rostros en imágenes, clasificación de rostros, entrenamiento y reconocimiento facial. Una vez que se genera esta secuencia se puede aplicar a diferentes objetivos, como por ejemplo identificador de emociones, reconocimiento de accesorios, como lentes, cubrebocas, etc.

3.1. Detección de rostros

El programa para llevar a cabo el reconocimiento o detección de rostros se lleva a cabo con las librerías de OpenCV cv2 y os. Lo que realiza el programa es primero declarar la ruta donde se encuentran las imágenes en una variable llamada imagesPath y la creación de una carpeta llamada "Rostros encontrados".

Para el proceso de la detección de los rostros se utilizó el dataset proveniente dentro del mismo OpenCV conocido como "haarcascade_frontalface_default.xml".

Finamente inicializamos un contador y creamos un ciclo for que realizará la lectura de las imágenes con *imread(),* para convertirla a escala de grises. Esto nos permitirá comparar la imagen con el data set y determinar si coincide para cada imagen dentro del directorio de imagesPathList.

Posteriormente creamos la variable faces la cual recibirá la información de la función *faceClassif.detectMultiScale()* dentro de la cual tenemos gray que es la imagen convertida a escala de grises, el factor de escala y el número de detección que puede haber cerca una con otras.

Para los rostros que detecte el programa dentro de la variable faces se dibujara un rectángulo en su rostro respectivo *cv2. Rectangles ()* con la imagen de entrada, las medidas del rectángulo que en este caso serán las coordenadas x & y donde se encuentren los rostros, el color y el tipo de borde que tendrá que en este caso es -1 que significa que será toda la figura.

Finalmente, los rostros encontrados se guardarán con la función cv2.imwrite() en la ruta especificada dentro del parámetro que es ['Rostros encontrados/rostro_{}.jpg'. format(count), rostro] donde {} será el contador que lleve el proceso de rostros del código:

```
import cv2
import os
imagesPath = "C:/Users/vioro/Downloads/DELFIN_PROJECT/4. Codigos/
3_Deteccion de rostros/Directorio de muestras"
imagesPathList = os.listdir(imagesPath)
print('ImagesPathList= ',imagesPathList  )

if not os.path.exists('Rostros encontrados'):
    print('Carpeta creada: Rostros encontrados')
    os.makedirs('Rostros encontrados')
faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface
_default.xml')
count = 0
for imageName in imagesPathList:
    #print('ImagenName= ',imageName)
    image = cv2.imread(imagesPath+'/'+imageName)
    #cv2.imshow('Imagen',image)
    imageAux = image.copy()
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    faces = faceClassif.detectMultiScale(gray, 1.1, 5)
    for (x,y,w,h) in faces:
        cv2.rectangle(image, (x,y),(x+w,y+h),(128,0,255),2)
    cv2.rectangle(image,(10,5),(450,25),(255,255,255),-1)
    cv2.rectangle(image,(10,5),(450,25),(255,255,255),-1)
    cv2.putText(image,'Presione R, para alamacenar los rostros
encontrados',(10,20), 2, 0.5,(128,0,255),1,cv2.LINE_AA)
    cv2.imshow('image',image)
    cv2.waitKey(0)
    k = cv2.waitKey(0)
    if  k == ord('R'):
```

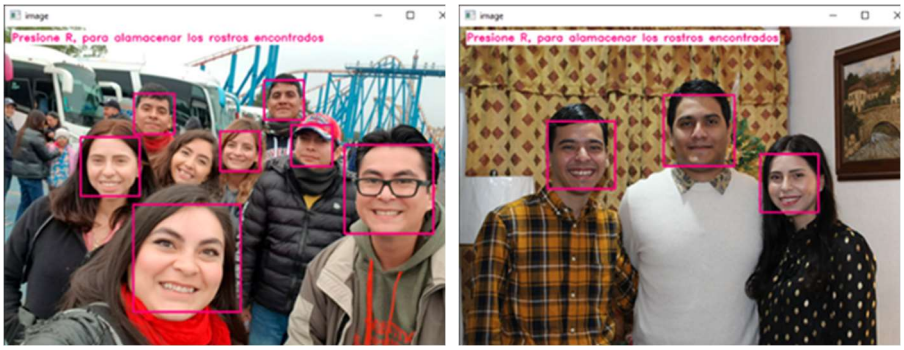


Fig.2. Programa para detección de rostros.

```
            for (x,y,w,h) in faces:
                rostro = imageAux[y:y+h,x:x+w]
                rostro = cv2.resize(rostro,(150,150), interpolati
            on=cv2.INTER_CUBIC)
                #cv2.imshow('rostro',rostro)
                #cv2.waitKey(0)
                cv2.imwrite('Rostros encontrados/rostro_{}.
            jpg'.format(count),rostro)
                count = count +1
        elif k == 27:
            break
    cv2.destroyAllWindows()
```

3.2. Clasificación de rostros

Para iniciar el reconocimiento facial por medio de visión artificial realizamos 3 programas; *Clasificador_Rostros.py* para crear nuestra data set que se utilizara y que proviene prácticamente del programa de detección de rostros anterior, el Entrenamiento.py para entrenar a la computadora y pueda identificar a la persona y por último *Reconocimiento_Facial.py* para la detección final.

El programa *Clasificador_Rostros.py* lo utilizaremos para que la cámara detecte a un usuario y lo almacene en una carpeta con su ID o identificador que en este caso es el nombre.

Utilizando además de las librerías cv2 y os utilizamos imutils. Primero, declaramos el nombre de la persona que queremos identificar en personalName. Ahora establecemos la ruta de la carpeta donde estarán todas las personas a identificar y una nueva variable llamada personPath que será la ruta establecida en dataPath y el nombre almacenado en personName.

Capturamos la información de la cámara dentro de la variable cap e inicializamos la variable faceClassif que utilizara el dataset para el reconocimiento de los rostros de haarcascades que antes se explicó.

Inicializamos el contador y con un ciclo **while** almacenamos la información de video dentro de la variable frame, la cual se modificará con la función de imutils para cambiar el tamaño del video y la conversión a escala de grises. Ahora utilizando el mismo código del programa anterior para detección de rostros:

```
import cv2
```

```
import os
import imutils
personName = 'Rodri'
dataPath = 'C:/Users/vioro/Downloads/DELFIN_PROJECT/4.
Codigos/Directorio de muestras II/Data'
personPath = dataPath + '/' + personName
if not os.path.exists(personPath):
    print('Carpeta creada: ',personPath)
    os.makedirs(personPath)
cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface
_default.xml')
count = 0
while True:
    ret, frame = cap.read()
    if ret == False: break
    frame =  imutils.resize(frame, width=640)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = frame.copy()
    faces = faceClassif.detectMultiScale(gray,1.3,5)
    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        rostro = auxFrame[y:y+h,x:x+w]
        rostro = cv2.resize(rostro,(150,150),interpolatio
       n=cv2.INTER_CUBIC)
        cv2.imwrite(personPath + '/rotro_{}.jpg'.format(c
       ount),rostro)
        count = count + 1
    cv2.imshow('frame',frame)
    k =  cv2.waitKey(1)
    if k == 27 or count >= 500:
        break
cap.release()
cv2.destroyAllWindows()
```

3.3. Entrenamiento de rostros

Ahora pasamos al programa de Entrenamiento.py que se encarga del entrenamiento de la visión artificial para la detección del usuario. En este programa tendremos 3 métodos distintos; Eigen Face, Fisher Face y LBPH Face.

Antes de continuar explicaremos un poco los métodos:

- El método de Eigen Faces emplea el método PCA (Principal Component Analysis), donde la idea principal es que un conjunto de datos de alta dimensión a menudo se describe mediante variables correlacionadas y, por lo tanto, solo unas pocas dimensiones significativas explican la mayor parte de la información.
- El método de Fisher Faces Este método es una mejora del anterior.

El método LBPH (Histogramas de Patrones Binarios Locales) presenta mejoras respecto a los métodos anteriores, ya que es más robusto ante cambios de iluminación. Además, dentro de la documentación de OpenCV la idea es no mirar la imagen completa como un vector de alta dimensión, sino describir solo las características locales de un objeto.

Es importante considerar que en cualquiera de los 3 métodos anteriores las imágenes de entrenamiento como de predicción deben estar en escala de grises, además de que en el método Eigenfaces se requiere que las imágenes de entrenamiento y prueba sean del mismo tamaño.

Enseguida se muestran los 3 programas mencionados:

```
ENTRENAMIENTO 1 (Eigen Face)
face_recognizer = cv2.face.EigenFaceRecognizer_create()
print("Procesando informacion...")
face_recognizer.train(facesData, np.array(labels))
face_recognizer.write('modeloEigenFace.xml')
print("Proceso finalizado...")
cv2.destroyAllWindows()

ENTRENAMIENTO 2 (Fisher Face)
face_recognizer = cv2.face.FisherFaceRecognizer_create()
print("Procesando informacion...")
face_recognizer.train(facesData, np.array(labels))
face_recognizer.write('modeloFisherFace.xml')
print("Proceso finalizado...")
cv2.destroyAllWindows()

ENTRENAMIENTO 3 (LBPH Face)
face_recognizer = cv2.face.LBPHFaceRecognizer_create()
print("Procesando informacion...")
face_recognizer.train(facesData, np.array(labels))
face_recognizer.write('modeloLBPHFace.xml')
print("Proceso finalizado...")
cv2.destroyAllWindows()
```

3.4. Reconocimiento facial

En este proceso ahora se explica el programa de reconocimiento, ya que tenemos el archivo entrenado que detectará a la persona podemos explicar el ultimo código (*Reconocimiento_Facial.py*).

Inicializamos la variable de face_recognizer con la función de *cv2.face.FisherFaceRecognizer_create()* y proseguimos a abrir el archivo entrenado xml con la función de *fase_recognizer.read()* con el nombre del archivo xml generado del programa anterior y empezamos a almacenar el video de la cámara en la variable cap, en seguida inicializamos la variable faceClassif que nos ayudara a identificar el rostro con el dataset de haarcascades. Continuamos con un ciclo while para la variable frame, con un cambio de escala. Iniciaclizamos la variable de la cara llamada faces con *faceClassif.detectMultiScale (gray, 1.3,5)* que detectara el rostro del video a escala de grises con un con sus parámetros establecidos para la detección.

Finalmente para cada faces obtenida mediante la función for la variable auxFrame pasara a la variable rostro con una reposicionamiento y un re-escalamiento a 150x150 pixeles con una interpolación simple cubica con la función antes explicada de *cv2.resize(),* con las imágenes generadas en el primer programa de detección y el video entrante utilizamos la función *fase_recognizer.preditc()* con la variable rostro dentro de su paréntesis para comparar las imágenes y buscar una coincidencia, si existe una o no se guardara en una variable llamad resultado en forma de un valor numérico. Aunque se pueden utilizar las 3 opciones de los métodos, utilizaremos el método de Fisher Face para comparar los resultados. Creamos una condición if para resultado comparado con 500:

```
face_recognizer = cv2.face.EigenFaceRecognizer_create()
face_recognizer.read('modeloEigenFace.xml')
cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface
_default.xml')
while True:
```

```
  ret,frame = cap.read()
  if ret == False:break
  gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
  auxFrame =gray.copy()
  faces = faceClassif.detectMultiScale(gray, 1.3,5)
  for (x,y,w,h) in faces:
    rostro = auxFrame[y:y+h,:x+w]
    rostro = cv2.resize(rostro,(150,150),interpolation =
cv2.INTER_CUBIC)
    result = face_recognizer.predict(rostro)
    cv2.putText
(frame,'{}'.format(result),(x,y+5),1,1.3,(0,255,120),1,cv2.LINE_AA)
    # EigenFaces
    if result[1] < 500:
        color 0 (0,255,120) if LABELS results [0]
    else:
        cv2.putText(frame,'Desconocido',(x,y-
20),2,0.8,(0,0,255),1,cv2.LINE_AA)
        cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),
       2)
  cv2.imshow('Video',frame)
  k = cv2.waitKey(1)
  if k== 27:
    break
cap.release()
cv2.destroyAllWindows()
```

3.5. Detección de mascarilla

Ahora para la variable de detección que es *mp_face_detection* utilizaremos una función de la librería media pipe que es *mp.solutions.face_detection*.

Definiremos un arreglo llamado labels los cuales tendrá los nombres de las carpeteas con las imágenes de dataset que son Con_Mascarilla y Sin_Mascarilla e inicializamos la variable que reconocerá este dataset que es face_mask con *cv2.face.LBPHFace Recognizer_create()* y *face_mask.read("face_mask_model.xml")* que es el método utilizado y el archivo generado, posteriormente dentro de la función *mp_face_detection.FaceDetection()* aplicamos un valor de confianza que será 0.5 para la variable face_detection.

Guardamos la información de la imagen del video de la cámara con frame.shape en dos variables que se llamaran height y width que almacenara el tamaño del video, inicializamos la variable llama results que obtendrá su resultado de la función *face_detection.process(frame_rgb)* que detectara si existe alguna coincidencia del video entrante convertido a RGB con la información entrenada en la carpeta data del dataset. Utilizando un condicional if para definir que si el *results.detections* tiene coincidencias obtendrá las coordenadas del rostro con la función *detection.location_data.relative_ bounding_box.[variable a pedir] *[ancho o alto (height o width)]*.

Finalmente se utilizara la función *face_mask.predict()* para la variable fase_image para detectar si existe alguna coincidencia positiva, este resultado lo guardara en la variable result. Los resultados solo podrán ser dos: “Con_mascarilla” o “Sin_mascarilla” que fue como se entrenó al programa.

```
mp_face_detection = mp.solutions.face_detection
#mp_drawing = mp.solutions.drawing_utils

LABELS = ["Con_Mascarilla", "Sin_Mascarilla"]
face_mask = cv2.face.LBPHFaceRecognizer_create()
face_mask.read("face_mask_model.xml")
```

```
cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
with mp_face_detection.FaceDetection(min_detection_confidence=0.5)
as face_detection:

    while True:
        ret,frame = cap.read()
        if ret == False: break
        frame= cv2.flip (frame,1)
        height, width, _ = frame. shape
        frame_rgb =  cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        results = face_detection.process(frame_rgb)
    # ---------------------------------------    Dibujo de los
resultados
        if results.detections is not None:
            for detection in results.detections:
xmin = int
(detection.location_data.relative_bounding_box.xmin*width)
ymin = int
(detection.location_data.relative_bounding_box.ymin*height)
w = int (detection.location_data.relative_bounding_box.width*width)
h = int
(detection.location_data.relative_bounding_box.height*height)
#cv2.rectangle(frame,(xmin,ymin),(xmin + w ,ymin + h),(0,200,120),2)
                if xmin < 0 and ymin < 0:
                    continue


face_image = frame [ymin: ymin + h , xmin : xmin + w]
face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2GRAY)
face_image = cv2.resize(face_image,(72,72),
interpolation=cv2.INTER_CUBIC)
                #cv2.imshow('Zoom',face_image)


                result = face_mask.predict(face_image)
                #cv2.putText (frame,'{}'.format(result),(xmin ,ymin
+ 5),1,1.3,(250,250,250),1,cv2.LINE_AA)


                if result[1] < 150:
                   color = (0, 255,120) if LABELS[result[0]] ==
"Con_Mascarilla" else (0, 0, 250)
cv2.putText(frame,'{}'.format(LABELS[result[0]]),(xmin, ymin - 25),
2, 1, color,1,cv2.LINE_AA)
cv2.rectangle(frame, (xmin,ymin),(xmin+w, ymin+h),color,2)


        cv2.imshow('Video',frame)
        k = cv2.waitKey(1) & 0xFF
        if k == 27:
            break


cap.release()
cv2.destroyAllWindows()
```

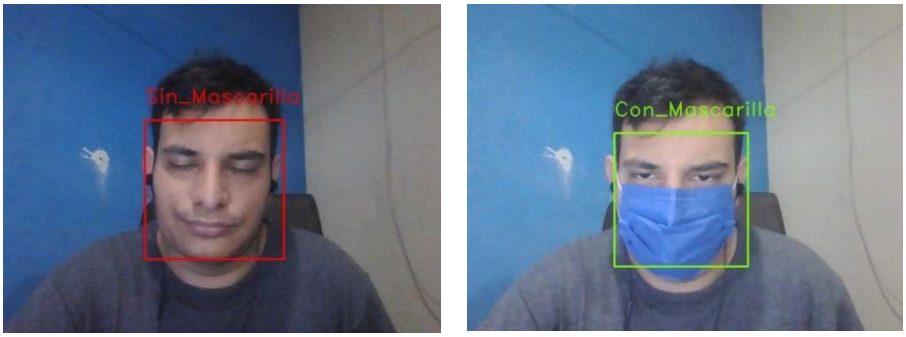


Fig. 3. Resultados del programa para la detección de la mascarilla.

En la figura 3 se muestran los resultados del programa para la detección de la mascarilla.

4. Resultados

En el caso del tiempo de entrenamiento para dos personas fue de aproximadamente 50 segundos aproximadamente con los primeros 2 métodos y de 25 segundos para el último método con un procesador AMD A-10 7860k (quad core a 4.0 GHz) y 16 RAM DDR3 1600MHz. Pero al incluir a una tercera persona los tiempos aumentaron a 5 minutos para los 2 dos métodos y a 1 minuto para el último método.

En cuanto a los resultados obtenidos por los 3 métodos utilizados el tercero de LBPH funciono de mejor manera siendo más estable demás de ser más rápido y fluido en comparación con los otros dos. Tanto el EigenFaces como Fisher Faces presentaban algunos falsos positivos mientas el otro no La precisión depende mucho de la iluminación que se tenga en la habitación y que refleja al usuario pen en general se obtuvieron resultados satisfactorios.

El resultado es muy satisfactorio y estable, además al haber utilizado un dataset ya hecho con muchas muestras con diferentes propiedades físicas podemos utilizar el programa con cualquier persona y de igual manera funcionara correctamente.

Referencias

1. Velavan, T. P., Meyer, C. G.: The COVID-19 epidemic. Tropical Medicine & International Health: TM & IH, vol. 25, no. 3, pp. 278–280 (2020)
2. Moreano, J. A. C., Pulloquinga, R. H. M., Lagla, G. A. F., Chisag, J. C. C., Pico, O. A. G.: Reconocimiento facial con base en imágenes. Boletâin Redipe, vol. 6, no. 5, pp. 143–151 (2017)
3. Gimeno-Hernández, R.: Estudio de técnicas de reconocimiento facial. Upc.Edu (2021) from https://upcommons.upc.edu/bitstream/handle/2099.1/9782/PFC_Roger Gimeno.pdf
4. Scarel, A., Martinez, G. M. D., Stegmayer, C. C. D. D., Muller, G. A. T.: Sistema de reconocimiento facial. Edu.ar (2021) http://sinc.unl.edu.ar/sincpublicatio ns/2010/SMS10/sinc_SMS10.pdf

5. Interpol: Reconocimiento facial (2021) https://www.interpol.int/es/Como-trabajamos/Policia-cientifica/Reconocimiento-facial
6. Chacua-Criollo, B. E.: Diseño de un sistema prototipo de reconocimiento facial para la identificación de personas en la facultad de ingeniería en ciencias aplicadas (FICA) de la Universidad Técnica del Norte utilizando técnicas de inteligencia artificial. Tesis de pregrado, Universidad Técnica del Norte. http://repositorio.utn.edu.ec/handle/123456789/9572 (2019)
7. Gualdrón, O. E., Suárez, O. M. D., Rojas, M. A. C.: Diseño de un sistema de reconocimiento de rostros mediante la hibridación de técnicas de reconocimiento de patrones, visión artificial e IA, enfocado a la seguridad e interacción robótica social. Mundo FESC, vol. 3, no. 6, pp. 16–28 (2013) http://www.revistapuce.edu.ec/index.php/revpuce/article/view/140/242
8. Porras, I. J. M.: Diseño de un sistema de reconocimiento facial como medio de control de acceso biométrico mediado por técnicas de inteligencia artificial como herramienta base de seguridad del CEAD IBAGUÉ. (2021) https://repository.unad.edu.co/bitstream/handle/10596/22930/Juan.Aldana.pdf?sequence=1&isAllowed=y

ISSN 1870-4069

Monitoreo de variables atmosféricas y percepción remota

Hermes Moreno-Álvarez[1], Juan Flores-García[2], Jesús Eduardo González[2], Carmen Santiago-Díaz[3], Gustavo Rubín-Linares[3], Catalina Rivera-Morales[4], Vanessa Baeza-Olivas[1], Xochitl de Jesus-Rojas[3]

[1] Universidad Autónoma de Chihuahua, Departamento de Ingeniería Aeroespacial, México

[2] Tecnológico Nacional de México, Instituto Tecnológico Superior de Poza Rica, México

[3] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, México

[4] Benemérita Universidad Autónoma de Puebla, Facultad de Ingeniería Química, México

{hm1713a, jesus95hidalgo}@gmail.com, {hmoreno, vbaeza}@uach.mx, juan.flores@itspozarica.edu.mx, {marycarmen.santiago, gustavo.rubin, maria.riveram}correo.buap.mx, xochitl.de@alumno.buap.mx

Resumen. En el presente trabajo se integran en una plataforma de bajo costo los diferentes subsistemas satelitales concluyendo la implementación y pruebas de funcionamiento. Una de las principales tareas es reportar la información proporcionada por los sensores que miden las variables ambientales en tiempo real, tales como la presión, altura, temperatura y humedad, de manera simultánea la información de la trayectoria que toma durante el ascenso y descenso de estos sistemas considerando una elevación mínima de 1000m con un dispositivo externo. El objetivo del diseño de este prototipo es ahorrar tiempo en la elaboración de prototipos espaciales educativos, en donde por este medio podrán verificar que sensores utilizar, observar el diseño electrónico, el microcontrolador, las plataformas y la programación utilizada para el desarrollo de prototipos futuros que ayuden al análisis de los datos atmosféricos. Sin embargo, también se pretende aportar en otras disciplinas con los diferentes subsistemas creados en el prototipo los cuales pueden tener diferentes aplicaciones.

Palabras clave: Subsistemas, satelital, sensores, ambiental, variables, programación, implementación, plataforma.

Hermes Moreno Álvarez, Juan Flores García, Jesús Eduardo González, et al.

Monitoring of atmospheric variables and remote sensing

Abstract. In the present work, the different satellite subsystems are integrated into a low-cost platform, concluding the implementation and performance tests. One of the main tasks is to report the information provided by the sensors that measure environmental variables in real time, such as pressure, height, temperature and humidity, simultaneously with information on the trajectory it takes during the ascent and descent of these systems considering a minimum elevation of 1000m with an external device. The objective of the design of this prototype is to save time in the elaboration of educational spatial prototypes, where by this means they will be able to verify which sensors to use, observe the electronic design, the microcontroller, the platforms and the programming used for the development of future prototypes that assist in the analysis of atmospheric data. However, it is also intended to contribute to other disciplines with the different subsystems created in the prototype, which may have different applications.

Keywords: Subsystems, satellite, sensors, environmental, variables, programming, implementation, platform.

1. Introducción

El uso de plataformas anfitrionas con sensores como parte de un sistema conformado por elementos que representan los subsistemas satelitales cada vez más intensivos en datos, impulsan la necesidad de una capacidad de enlace descendente de gastos de mayor velocidad y eficiencia energética.

Los CubeSats, el estándar dominante para los nanosatélites, tienen un tamaño, peso y potencia limitados, lo que dificulta la instalación de antenas de radiofrecuencia (RF) de alta ganancia. Esto empuja a la mayoría de las misiones de CubeSat con necesidades de alta velocidad de datos a utilizar estaciones terrestres con grandes aperturas de alta ganancia (diámetros de plato que oscilan entre 5 y 20 m).

También existen desafíos regulatorios para obtener licencias de RF con un ancho de banda sustancial para misiones CubeSat; incluso la gestión de las numerosas solicitudes de licencias de ancho de banda estrecho se ha convertido en un problema. Una alternativa a las comunicaciones de RF tradicionales es deseable para CubeSats.

Actualmente las cosas han cambiado, ya que conforme ha avanzado la tecnología, y la estandarización, ha bajado el costo de los componentes utilizados para este tipo de misiones, dando oportunidad a estudiantes y empresas que con un menor presupuesto tienen el alcance de realizar sus propias investigaciones [1].

Una alternativa que ha ayudado a realizar misiones espaciales, sobre todo a los estudiantes, son los llamados Cubesat, se trata de sistemas que representan los elementos que componen un satélite de pequeñas dimensiones, son más baratos en cuanto a materiales, su construcción y sus lanzamientos [2]. Debido a sus características como su peso, su geometría y carga útil, estos dispositivos suelen ser menos complejos

que los satélites comerciales. Sin embargo, eso no los exime de tener un gran potencial y un gran alcance dentro de las investigaciones [4].

Este proyecto se enfoca en realizar un prototipo de un nanosatélite con un presupuesto bajo que tenga la capacidad de comunicación bidireccional con un centro de control en el cual se puedan monitorear algunas variables de la atmósfera. Se puedan analizar las variables y gestionar los datos obtenidos. A través de esta información se podría obtener un análisis concreto acerca de algunas variables atmosféricas.

Para lograr este cometido, primero se tendrá que seleccionar y probar cada uno de los elementos, integrarlos con un microcontrolador para obtener el conjunto de variables que se requieren, para posteriormente realizar la comunicación bidireccional del prototipo con una estación que tenga acceso en tiempo real de los datos que se quieren obtener. Finalmente, se obtendrán resultados de su capacidad en la obtención de datos, visualización de los gráficos de los datos obtenidos, tiempo de transmisión y costos.

2. Marco teórico

Subsistemas que conforman los satélites "Un satélite está dividido principalmente en dos partes, la plataforma y la carga útil, la carga útil es la encargada de realizar la misión del satélite y la plataforma es la encargada de darle soporte al funcionamiento del satélite, como energía, posicionamiento, comunicaciones, trabajo a determinados niveles de temperatura, entre otros.

De este modo la plataforma, es el conjunto de subsistemas que dependen uno del otro para lograr el correcto funcionamiento del satélite" [7]. El subsistema de control de orientación es un sistema muy importante para el satélite, consta de componentes que le ayudan a corregir su posición, en dado caso de que alguna antena no esté apuntando para el lugar correcto.

Esta corrección se realiza mediante propulsores utilizando un sensor giroscopio como referencia o un GPS. El satélite puede corregir su posición mediante algoritmos preestablecidos, el cual le da un alto grado de autonomía.

Para el subsistema de energía está constituido por paneles solares como la fuente principal de energía, además de que cuenta con baterías que almacenan la energía que se obtiene de los paneles y sirven para dará inicio a las operaciones un instante después de ser lanzado del vehículo que lo transporta antes de que los paneles solares empiecen a alimentar al satélite.

Las baterías reabastecen cuando no hay suficientes rayos del sol para alimentar todo el satélite. En el caso del *subsistema de telemetría* se da a conocer el estado de los demás subsistemas, en donde mediante sensores se miden variables vitales del satélite como es: voltaje, corriente, temperatura, presión, entre otras variables más. Estas mediciones se envían a través de canal de comunicación espacial para informar a la base lo que ocurre con el satélite.

Otro de los subsistemas a considerar es el de telemetría y comando, este nos permite mandar instrucciones desde la base de control a los subsistemas del satélite, a través de una línea de comunicación, las instrucciones son codificadas para que solo la base tenga el control del satélite y ningún otro equipo pueda acceder al control del satélite.

```
#define SSID "REDWIFISAT"
#define SSID_PASSWORD "PassWordSat"

#define USERNAME "Usuario_sat"
#define DEVICE_ID "prototipo"
#define DEVICE_CREDENTIAL "163928"
```

Fig. 3. Líneas de código de configuración de usuario, ID del dispositivo, credenciales del dispositivo y la configuración de la red.

```
thing["dht22"] >> [](pson& out){
out["Temperatura"] = dht.readTemperature();
out["Humedad"] = dht.readHumidity();
out["Temperature"] = bmp.readTemperature();
out["Presion"] = bmp.readPressure();
out["Altitud"] = bmp.readAltitude(1013.25);
out["AnguloX"] = AcX;
out["AnguloY"] = AcY;
out["Latitud"] = gps.location.lat();
out["Longitud"] = gps.location.lng();
```

Fig. 4. Líneas de código de la configuración de envío de variables a la plataforma.

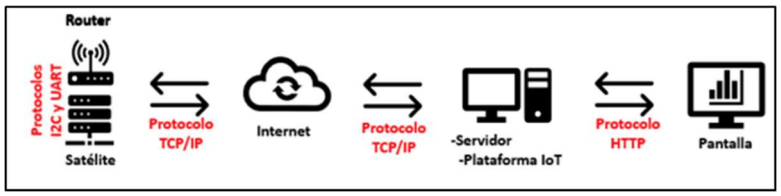


Fig. 5. Esquema de interacción de protocolos para la trasferencia de datos.

En cuanto al subsistema de control térmico podemos decir que controla las temperaturas dentro de la nave, para que los componentes y sistemas trabajen en óptimo funcionamiento, ya que las temperaturas pueden ser muy extremas en el espacio exterior, sobre todo cuando hay eclipse o cuando los rayos del sol no pegan directamente al satélite. Para el control térmico, se emplean conductores de calor, radiadores y calefactores eléctricos.

3. Estado del arte

Los modelos denominados CanSats se han utilizado principalmente en los niveles de educación superior y especialidad, utilizando una amplia variedad de hardware y software, con algunos intentos de estandarización a través de "kits" [2, 3, 4]. Esto debido a una notable falta de material educativo que pueda ser utilizado para crear clases donde pueda aprovecharse esta tecnología, pues los modelos reales implican electrónica de grado espacial con costos que sobrepasan las posibilidades de los centros educativos.

Los cubesat en su presentación de simuladores de mesa, así como los modelos reales puestos en órbita como lo fue el AztechSat-1 se han utilizado principalmente a nivel

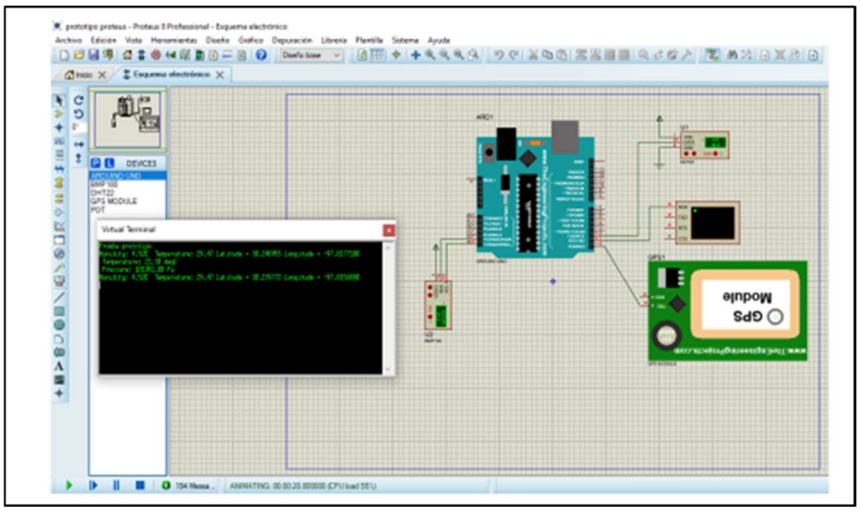


Fig. 6. Simulación de adquisición de datos a través de Proteus y la salida en la terminal virtual.

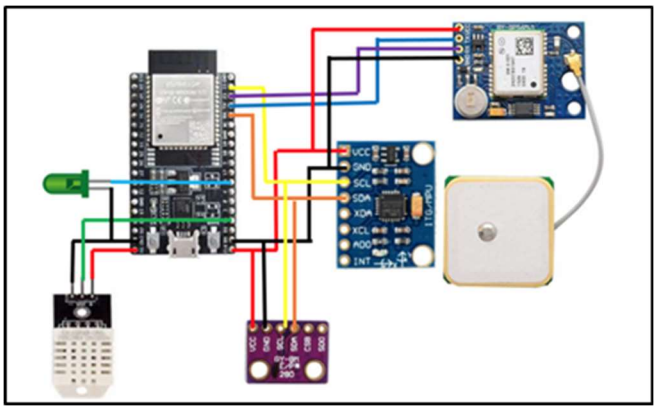


Fig. 7. Integración del circuito.

universitario teniendo una variada participación de disciplinas las cuales en mayor o menor grado están relacionadas con la tecnología espacial. Por su lado los cansats pueden incluso cubrir un sector más amplio en los niveles educativos que pueden empezar a nivel de media y media superior.

Los satélites en general poseen una estructura similar y con los cansats. Se pretende reproducir el funcionamiento de éstos subsistemas, sin embargo, las tecnologías que se utilizan son variadas esto básicamente dependiendo de los objetivos o misiones. Es importante mencionar que las aplicaciones de éstos subsistemas pueden extrapolarse a otras áreas, quienes pueden aprovechar ésta tecnología tanto en la parte de sensado como de transmisión y almacenamiento de datos.

El dispositivo que se presenta proporciona la capacidad de medir in situ diferentes variables atmosféricas, las cuales son integradas en el dispositivo, documentando en

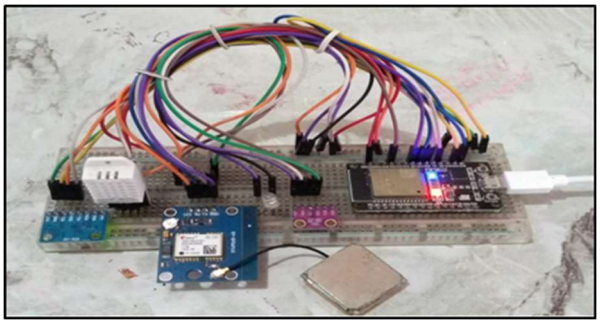


Fig. 8. Prototipo armado en funcionamiento.

tiempo real los parámetros obtenidos con un volumen considerable de datos almacenados para su manipulación y bajo costo. Se realiza una descripción de los diferentes elementos utilizados en el dispositivo y el diseño del mismo tomando en cuenta los requisitos de medición. Los resultados presentados ilustran la calidad de los datos proporcionados por las mediciones realizadas de manera simultánea.

4. Desarrollo

Los componentes son piezas fundamentales dentro de cada dispositivo y es importante seleccionar minuciosamente cada uno de ellos. En el caso de los sensores, puede variar el rango de medición o el voltaje de operación entre un sensor y otro, motivo por el cual, se verifica cada una de las características de los elementos que se van a integrar. Enumerando a continuación algunas de las características de los componentes a utilizar en este prototipo.

4.1. Sensor GY-521 MPU6050

Tiene todo lo necesario para medir movimiento en 6 grados de libertad, combinando un giroscopio de 3 ejes y un acelerómetro de 3 ejes. Trabaja con un voltaje de operación de 3 a 5 V. Puede medir las aceleraciones en los rangos de 2g/4g/8g/16g, además cuenta con giroscopio, cuyos rangos son: 250Grad/Seg, 500Grad/Seg, 1000Grad/Seg, 2000Grad/Seg, con sensibilidad de giroscopio de 131 LSBs/dps, interface de I2C, conversor AD de 16 bits (salida digital). Sus dimensiones son de 19x15x5 mm y tiene un peso de 15 gramos.

4.2. Cámara ESP32 CAM módulo

Esta cámara opera con un voltaje de 5 volts, cuenta con módulo de wi-fi 802.11b/g/n, con un tipo de cámara OV2640 2MP, cuenta con CPU de 32 bits de doble núcleo de baja potencia, frecuencia principal de 240 MHz, potencia informática de hasta 600 DMIPS, velocidad de reloj de hasta 160 MHz. Incorpora SRAM 520 Kb, 4MPSRAM externa, soporta interfaces UART/SPI/I2C/PWM/ADC/DAC, soporta cámaras OV2640/OV7670, flash incorporado, soporta tarjetas TF micro SD (máximo 4 GB),

Fig. 9. Pantalla de visualización de las variables atmosféricas humedad, temperatura, presión barométrica, altitud, nivel en el eje x y y, medición del giroscopio, sin unidades.

Fig. 10. Pantalla de visualización que muestra el comportamiento de las variables, temperatura, altitud, presión, humedad y nivel del prototipo con respecto al tiempo y la ubicación en tiempo real.

compatible con modos de operación STA/AP/STA+AP, con antena PCB, cuenta con conectores u.FL y FPC. Sus dimensiones son de 27x40.5x4.5 mm y tiene un peso de 20 gramos. En la figura se muestra el aspecto físico del sensor con sus entradas y salidas del mismo.

5. Prueba de funcionamiento de componentes

Se realizaron las pruebas con los sensores a traves del gestor del IDE de Arduino, verificando la funcionalidad de cada uno de los sensores, considerando que en el prototipo final se utilizará el microcontrolador ESP32.

5.1. Programación

Una vez integrada la programación se cargó el código de todos los sensores en el microcontrolador ESP32, para corroborar el correcto funcionamiento del código los datos obtenidos de esta prueba se mostraban en la consola del IDE de Arduino.

Para programar el código de la comunicación se configuró la red wifi y los parámetros como el nombre del usuario, ID del dispositivo y credenciales para la comunicación con la plataforma IoT Thinger.io, como se muestra en la figura 3.

Después de unir el código de comunicación con el código que integran los sensores, se agrega el código que indica que datos se envían, como se muestra en la figura 4.

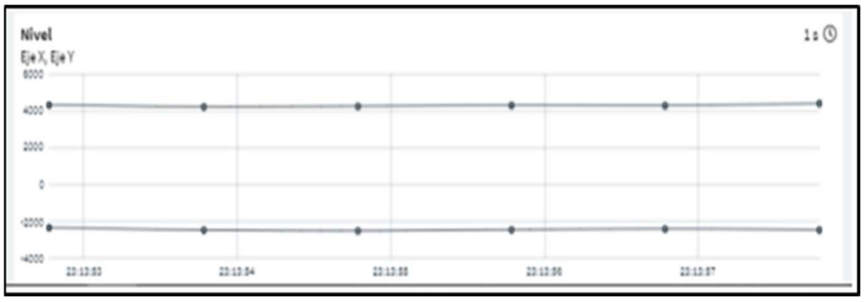


Fig. 11. Muestra el comportamiento del eje x y eje y con respecto al tiempo, cuando el prototipo se encuentra nivelado.

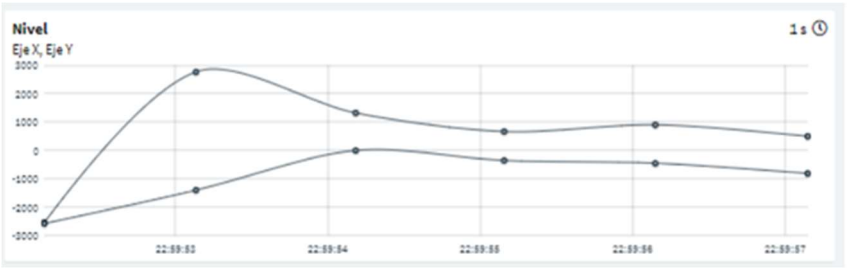


Fig. 12. Muestra el comportamiento del eje x y eje y con respecto al tiempo, cuando el prototipo ha sufrido alguna inclinación.

5.2. Comunicación

Se configura el dashboard con la opción From Device Resource, para indicar que la fuente del valor viene de un dispositivo, el nombre del dispositivo, el nombre del grupo de variables, la variable y el tiempo con el cual se estaría actualizando la información.

En la figura 5 se muestra cómo interactúan los protocolos de comunicación, el protocolo I2C fue utilizado para obtener los datos del sensor giroscopio y del sensor BMP280, y el protocolo UART se utilizó para obtener los datos del sensor GPS. Una vez recopilados los datos en el microcontrolador se utilizó el protocolo TCP/IP para mandar todos estos datos a través de internet a la plataforma IoT,

Thinger.io, y posteriormente se utilizó el protocolo HTTP para mostrar los datos a través de gráfico mejor conocidos como dashboards

6. Resultados

En la figura 6 se muestra el resultado del circuito de la simulación en el software Proteus, en donde se puede observar el sensor de temperatura, módulo de GPS y al sensor de presión y altura conectados de manera virtual a un microcontrolador Arduino UNO, que a su vez está conectada a una terminal virtual para mostrar los datos obtenidos de los sensores.

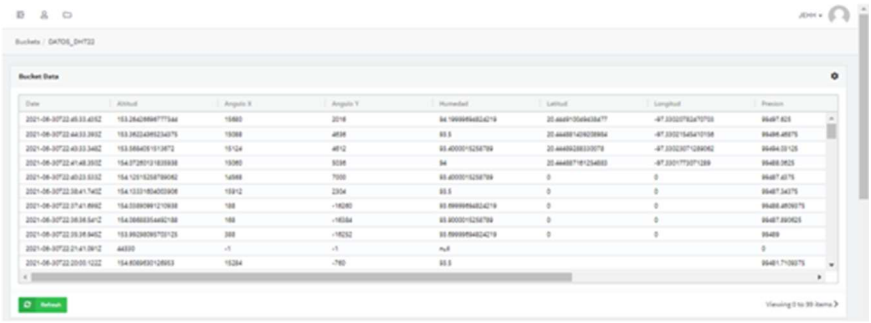


Fig. 13. Base de datos con los registros de las variables obtenidos a través de los sensores.

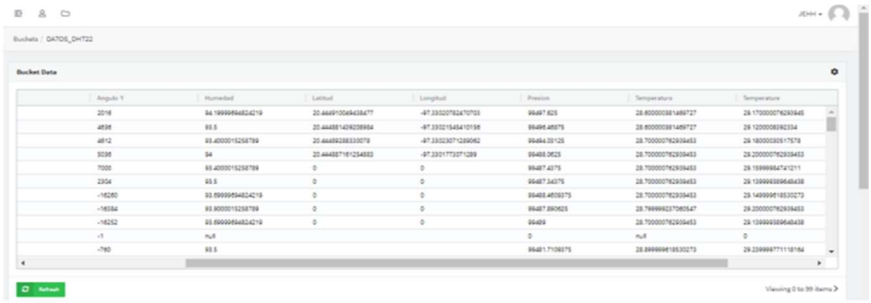


Fig. 14. Continuación de la base de datos obtenida de la plataforma Thinge.io.

En la figura 7 se muestra el esquema resultante del circuito del prototipo, realizando la integración de los componentes, respetando las entradas de los protocolos de comunicación como fue el caso de los sensores BMP280, GY-521 MPU6050 y el módulo de GPS. Todo el circuito fue alimentado con 3.3 volts que son los valores de desempeño del microcontrolador ESP32.

En la figura 8 se muestra la integración de los componentes para realizar las pruebas. En lo que respecta a la velocidad de la comunicación se obtuvieron resultados muy favorables, sobre todo con la rapidez y fluidez en la transmisión de los datos del microcontrolador con la plataforma Thinger.io, realizamos una prueba, la cual consistía en cronometrar el tiempo en que tardaba en encenderse el led, accionando el botón desde los gráficos, y se obtuvo un tiempo de 0.40 segundos con una velocidad de carga y descarga en internet de 5.03 Mpbs y 8.66 Mbps respectivamente, por lo que la latencia puede variar dependiendo la velocidad del internet.

En la figura 9 se observa gráficamente el resultado de las variables interactuando en tiempo real con las opciones monitoreo y estadística, en el primer caso se muestra la variable de Humedad obtenida por el sensor DHT22, con la unidad de porcentaje en un gráfico llamado Gauge, a un lado se encuentra la variable de temperatura obtenida por el sensor DHT22, con la unidad de centígrados en un gráfico llamado Tachometer, en el cual se configuraron alarmas en el color del grafico de acuerdo a un rango en la escala establecido, también se muestra la variable de temperatura obtenida por el sensor BMP280, con la unidad de centígrados en un gráfico llamado Gauge también con alarmas, enseguida la variable de presión proveniente del sensor BMP280 con la unidad

bar en un gráfico llamado Progressbar, finalmente la variable altitud obtenida por el sensor BMP280, en metros y en un gráfico llamado Tachometer, se muestran las variables de nivel del eje x y eje y, las cuales no tienen ninguna unidad en unos gráficos llamados Text/Value, los cuales cuentan con tres alarmas, cambian de color, verde o rojo dependiendo si está o no equilibrado el eje respectivo.

La figura 10 muestra las *Estadísticas* que genera la plataforma, en donde se puede ver el comportamiento de las variables de temperatura, altitud, presión, humedad y nivel en los ejes X y Y, con respecto al tiempo, las cuales se van registrando gráficamente cada 5 segundos, así como la ubicación en tiempo real a través de un mapa, gracias a las coordenadas de latitud y longitud obtenidas del módulo de GPS.

En la figura 11 se puede observar el comportamiento del eje x y eje y con respecto al tiempo, cuando los dos ejes se encuentran nivelados, se puede apreciar que la gráfica se mantiene estable en comparación con la figura 12 la cual muestra cuando el eje x y eje y, tienen alguna inclinación o movimiento, se ve como los puntos graficados sufren un cambio significativo. Por lo que este tipo de gráficas son ideales para identificar cualquier anomalía que suceda en el prototipo.

Las figuras 11 y 12 muestran los resultados de la base de datos, que en este caso se programó para almacenar los datos de todas las variables cada minuto la cual solo puede tener 100 registros almacenados y posteriormente se va eliminando el último. En la figura 13 se muestra la base de datos con 8 columnas etiquetadas como *Date* para la fecha y hora, enseguida las correspondientes a las variables Altitud, Ángulo x, Ángulo y, Humedad, Latitud, Longitud y Presión. Finalmente, en la figura 14 se muestra la continuidad de la base de datos en las columnas con el nombre de las variables faltantes, las cuales son las dos variables de Temperatura obtenidas.

Se compararon las mediciones de los sensores en el prototipo físico con las que se midieron en la plataforma obteniéndose precisiones menores al 0.05%.

7. Conclusiones

Se puede hacer las siguientes conculsiones.

1. En este prototipo se integraron diferentes subsistemas electrónicos, que permitieron recopilar la información de variables atmosféricas en tiempo real a una altura de mil metros sobre el nivel del mar para simular una misión satelital de comunicación y percepción remota.
2. Se diseñó un prototipo con los subsistemas mínimos requeridos considerando las limitaciones en peso, dimensiones, transmisión de información y la creación de las bases de datos a través del software Proteus y la plataforma Thinger.io que sirvieron de apoyo para integrar dichas soluciones.
3. Se establecieron las variables atmosféricas que se miden en el prototipo, en donde se determinó que las variables de temperatura, humedad, altura, localización y el nivel en los ejes X y Y son indispensables para vigilar el comportamiento de la atmósfera.
4. Se seleccionó el microcontrolador ESP32, se debido a que tiene integrado un módulo wifi, además de que es compatible con la IDE de Arduino y sobre todo porque es compatible con la plataforma de IoT que se eligió para realizar la comunicación.

5. Se configuró la comunicación entre los diferentes protocolos de comunicaciones que los diferentes subsistemas requerían.
6. Se seleccionó la plataforma Thinger.io para una comunicación y visualización de los datos obtenidos en tiempo real, así como también el almacenamiento en la base de datos que la misma plataforma proporciona.
7. Este prototipo electrónico se convierte en una herramienta útil, no solo para el sector aeroespacial sino para múltiples áreas en donde se les puede obtener aplicaciones para el monitoreo de diferentes variables, ocupando como base información expuesta en este documento las cuales facilita una alternativa más clara de telemetría.

Referencias

1. Orts-Torres, C.: Propuesta y diseño de un nano-satélite de comunicaciones para fines educativos (2021)
2. Wertz, J. R., Larson, W. J.: Space mission analysis and design. Springer Netherlands (1999)
3. Cubesat project: Cubesat Design Specification (CDS) (2013)
4. Opazo-Toro, T. I.: Requerimientos, implementación y verificación del nano-satélite Suchai (2013)
5. Chin, J., Coelho, R., Foley, J., Johnstone, A., Nugent, R., Pignatelli, D., Pignatelli, S., Powell, N., Puig-Suari, J.: Basic concepts and processes for first-time cubesat developers. NASA, pp. 96 (2017)
6. Toorian, A.: CubeSat Design Specification (2015)
7. Bohorquez Garzón, Y. S.: Diseño conceptual y preliminar de un CubeSat de bajo costo (2019)
8. Jiménez-Fernández, C. J., López-Ojeda, A., León de Mora, C.: Metodología de diseño electrónico dentro de prácticas obligatorias de laboratorio, no. 37, pp. 19–27 (2010)
9. Barrón, M.: Uso didáctico del software de ayuda al diseño electrónico PROTEUS (1970)
10. Corona, A. E.: Protocolos TCP/IP de internet (2004)
11. Luna del Águila, F.: Implementación del protocolo HTTP paralelizado en cliente y servidor (1999)
12. Sánchez-Martínez, I.: Empleo de bus 12C en el microcontrolador PIC 16F73. Universidad Central Marta Abreu de Las Villas (2008)
13. Lozano-Cruz, D., López-Jiménez, J. A., Cruz-Avilés, D., Granillo-Macías, R.: Diseño de una red de distribución mediante datos obtenidos de una interfaz de programación de aplicaciones. vol. 7, no. 13, pp. 42–48 (2020)
14. Monge De La Cruz, L. A., Torres-Herrera, J. P., López Chico, L. E., Navarro Cota, C. X.: Análisis comparativo de servidores de mapas. no. 10, pp. 1–10 (2010)

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación