EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 152 No. 11
November 2023

Research in Computing Science

Research in Computing Science, Año 22, Volumen 152, No. 11, noviembre de 2023, 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 2023.

Research in Computing Science, year 22, Volume 152, No. 11, November 2023, is published monthly by the Center for Computing Research of IPN.

Volume 152(11)

Advances in Artificial Intelligence

Lourdes Martínez Villaseñor (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2023

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

Behavior's Study of some Classic SVD-Models with Noisy Data in Movie Recommender Systems

Cupertino Lucero-Álvarez[1,2], Perfecto Malaquías Quintero-Flores[2], Edgar Moyotl-Hernández[3], Carlos Artemio Ortiz-Ramírez[1], Patricia Mendoza-Crisóstomo[1], Maria Vázquez-Vázquez[1]

[1] Universidad Tecnológica de Izúcar de Matamoros, Tecnologías de la Información, Mexico

[2] Universidad Autónoma de Tlaxcala, Facultad de Ciencias Básicas Ingeniería y Tecnología, Mexico

[3] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias Físico Matemáticas, Mexico

{clucero, cortiz, pmendoza, mari.vazquez}@utim.edu.mx, parfait.phd@gmail.com, moyotl@buap.edu.mx

Abstract. This paper presents a study about the behavior of three variants of the SVD algorithm in Collaborative Recommender Systems (CRS). For this, two MovieLens DataSets are used, and five variants in each DataSet with different degrees of randomness. Specifically, a comparison of the classic models is presented: Funk-SVD, Regularized-SVD, and Bias-SVD. The underlying idea is to observe that, as the degree of randomness in the data increases, the precision of the recommendations decreases, and the hidden relationships that may exist in the original data they get lost because of the noise. For this, we have configured two groups of experiments: in the first group, in each execution 10, 20 and 30 Latent Factors (LFs) were considered in the three models, while in the second group from 5 to 80 LFs were used in the regularized-SVD model. The prediction error was minimized using the MSE (Mean Square Error) metric and the ADAM optimizer. The results show that SVD with biases performs better, under the conditions of these experiments, and that noise affects the hidden relationships between the data.

Keywords: Collaborative recommendation systems, matrix factorization, singular value decomposition, latent factors, noisy data.

1 Introduction

Due to the vigorous growth of electronic commerce today, the need for efficient management of the Big Data generated is pressing. In the area of Recommendation Systems (RS), applications need efficient algorithms in the use of the computational

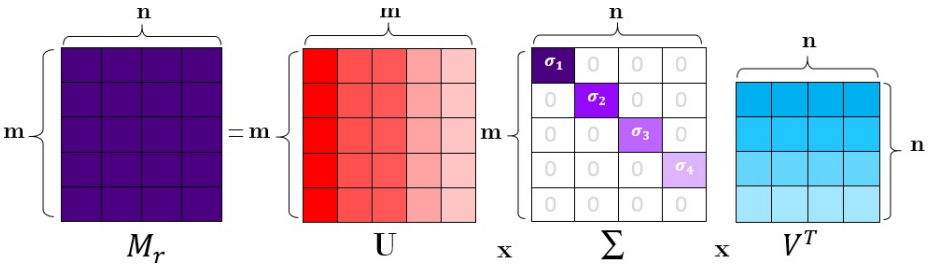


Fig. 1. Matrix factorization.

resources, and that offer quality recommendations, based on this, the RS filter the information according to the user profiles, and predict the elements of interest for each user in a personalized way. The elements can be books, websites, movies, tourist routes, hotels, e-learning materials, e-commerce articles, to name a few [2, 20].

In Collaborative Filtering (CF) recommendations are made based on the tastes of the active user's neighbors [15], and the ratings are recorded in a data structure called Ratings Matrix M_r [1, 14].

Early approaches used full M_r and faced bottlenecks due to dimensions and large spread of data, later, more successful approaches reduce the dimensions of the data through Matrix Factorization (MF) techniques, in whose decomposition the loss of information is not considerable.

In linear algebra, MF consists of decomposing the matrix as a product of two or more matrices according to a canonical form [23], in such a way that it is easier to work with them, in [18] a pioneering investigation is presented in the use of the technique, in which SVD is compared whith traditional filtering of memory-based CRS, and SVD is validated. The main problems of CRS, such as data sparseness and cold start, have been widely addressed [8, 22, 21], but little has been studied about the behavior of algorithms with respect to biased, noisy or completely distorted data.

Addressing the problem of noisy or biased data can help generate mechanisms that improve recommendations, when users face biases due to overexposure, and popularity, which can confuse their preferences and give high ratings or "likes" to articles that are not of their interest [10]. Although biases have already been addressed from different perspectives, such as those that consider the degree of dispersion with respect to the mean, or biases due to context or temporality.

We believe that investigating the behavior of SVD models in relation to noisy data could help decide on their use with respect to the applications domain. In this work, the behavior of three classic SVD algorithms is studied with respect to noisy data, for which five DataSets were generated with a certain degree of randomness in two MovieLens DataSets, and two groups of experiments with different number of LFs were configured.

We have mainly relied on the research reported by Koren [11] to describe the SVD models. The rest of this work is organized as follows: Section 2 addresses the theory of the implemented SVD models. Section 3 describes the experimental work and results. Finally, in section 4 conclusions are made and future work is presented.

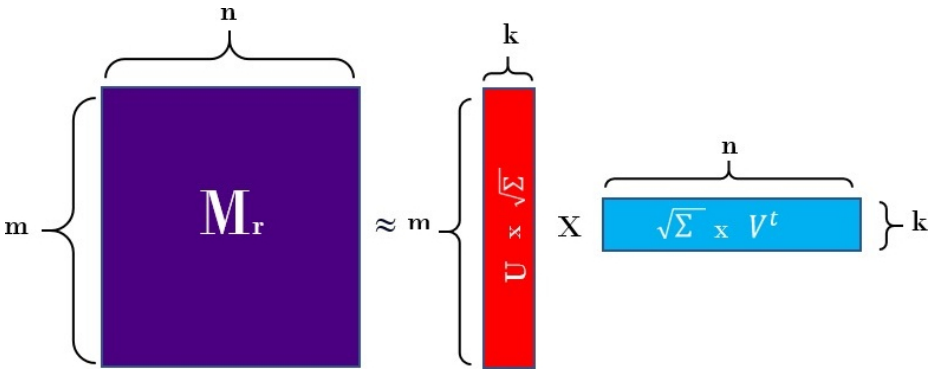


Fig. 2. Funk-SVD.

2 Theory About the SVD Method

SVD has been used to describe data in a reduced representation of its key characteristics, for example in megapixel image processing [5], high-resolution videos [13], natural language processing tasks [12], and recommendation systems [1]. SVD is used in Big Data, for example in Principal Component Analysis (PCA) to identify dominant patterns and correlations [3].

SVD is also used by big websites like FaceBook for their friend recommendations, Google for page ranking, Amazon and Netflix for their RS. In MF, a result of numerical linear algebra states that every matrix A can be represented as: $A = U \sum V^T$, where U and V are unitary matrices, in [16] the proof of the decomposition theorem can be found. Now, $\sum$ is a non-negative diagonal matrix and is arranged in descending order with respect to its magnitude $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_n$ all positive, and the last ones can be zero; which means that the first column of U corresponds to σ_1 just like the first column of V, and the second column of U corresponds to σ_2 like the second column of V, and so on, so this hierarchy indicates the order of importance they have in the decomposition of A. As shown in Fig. 1 for M_r. In SVD, the physical interpretation of the columns of U and V is intuitive, as are the values of the diagonal of $\sum$.

In the context of a CRS of movies, M_r is a matrix of column vectors, which contain the level of liking of the users who have rated the elements, each column vector contains the "likes" of each user who has qualified the movie that represents the vector, in this sense, the columns of U would be the eigen-movies, and they would be arranged in order of importance with respect to their ability to describe the columns of M_r, that is: the movie represented by the first eigenvector would be intuitively more relevant than the movie represented by the second eigenvector, and so on. According to the Eckart-Young theorem [4, 7], the best approximation of A is obtained by considering only the largest k singular values and setting the smallest to zero:

$$A_{n\times m} \approx U_{n\times k} \times \sum_{k\times k} \times V^T_{k\times m}. \tag{1}$$

Table 1. DataSets used in the experiments.

DataSet	Ranks	Us	Movies	Ratings	Density
Small (DS1)	[0.5:5] With half star increments	610	9,724.00	100,836.00	1.7%
Ratings100K (DS2)	[1:5] In one star increments	943	1,682.00	100,000.00	6.3%

It is possible to simplify the SVD process by obtaining only the user and element factors [17], decomposing the matrix $\sum$ into two equal matrices, as shown in Fig. 2. In SR, the reduction of the dimensions of M_r is fast as long as the matrix is dense, In most cases, 10% of the largest singular values and the corresponding vectors of the matrices U and V are sufficient to represent 98% of the total elements of M_r to a good approximation, which is done by the inner product of the vectors of the SVD decomposition. However, the high dispersion of data is always a problem to be solved.

2.1 Model: Funk-SVD

In movie RS, Funk proposed the decomposition of M_r into the matrices U and V considering that $\sum$ has been multiplied in either of them implicitly, as shown in Fig. 2, and in this way reduced the dimensions of the data [19]. Funk starts by filling the matrices U and V randomly, and then uses ML to modify the inputs to get a good approximation of M_r. A score of M_r can be predicted using the equation 2:

$$\hat{r}_{u,i} = \sum_{k=1}^{K} U_{u,k} \times V_{k,i}, \tag{2}$$

where K is the number of LFs. The error is the difference between the actual value and the predicted value: $E = M_{u,i} - \hat{r}_{u,i}$, and MSE is used to calculate the total error. The idea is that user u's final rating on item i can be estimated by adding user u's interest in i on each dimension of the hidden feature k, equation 3:

$$E = \sum_{u} \sum_{i} \frac{1}{2} (M_{u,i} - \hat{r}_{u,i})^2. \tag{3}$$

The objective is to minimize E with respect to U and V, using some optimizer such as Stochastic Gradient Descent (SGD). Even though M_r is very sparse, the algorithm works because it only takes the known inputs.

2.2 Model: Regularized-SVD

In this model, taken from [11, 19], the learning of p_u and q_i is achieved by minimizing the regularized quadratic error, as in the equation 4:

$$\min_{q^*,p^*} \sum_{(u,i)\in K} (r_{u,i} - q_i^t \cdot p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2), \tag{4}$$

where K is the set of pairs (u, i) for which $r_{u,i}$ is known, the constant λ controls the degree of regularization and is usually determined by cross-validation [11]. To minimize the equation 4 an optimizer such as SGD is used.

Table 2. Variants of MovieLeans DataSets.

Exp	Data	Description
A	**DataSet 1**	Actual data, no change.
B	**DataSet 2**	Random ratings on rated movies, and same frequency distribution of ratings.
C	**DataSet 3**	Random ratings on rated movies, and random frequency distribution of ratings.
D	**DataSet 4**	Random ratings on rated movies, and same frequency distribution of ratings but with random assignment of ratings to rated movies by each user.
E	**DataSet 5**	Random ratings on rated movies, and random frequency distribution of ratings, but with random assignment of ratings to movies rated by each user.
F	**DataSet 6**	Random ratings on rated movies, and random frequency distribution of the ratings, but with random assignment of the ratings to the movies in the data set.

The algorithm goes through all the ratings in the training set, calculating in each case $r_{u,i}$ and its associated prediction error, as shown in the equation 5:

$$e_{u,i} = r_{u,i} - q_i^t \cdot p_u. \tag{5}$$

The parameters are then changed by a magnitude proportional to γ in the direction opposite to the gradient. As in the equation 6:

$$q_i \leftarrow q_i + \gamma \cdot (e_{u,i} \cdot p_u - \lambda \cdot q_i), p_u \leftarrow p_u + \gamma \cdot (e_{u,i} \cdot q_i - \lambda \cdot p_u). \tag{6}$$

There are other methods that can also be used in the ML process, such as Alternating Least Squares (ALS), especially in implicit feedback [11].

2.3 Model: Bias-SVD

This model considers the biases [11] related to the deviation that each rating has with respect to the averages of the active user and element, and is compared with the global average. Thus, $b_{u,i} = \mu + b_i + b_u$, where μ is the global average in M_r, and the parameters b_i and b_u are the observed deviations of user u and item i respectively. Therefore, to estimate the rating of user u for element i, we have:

$$\hat{r}_{u,i} = \mu + b_i + b_u + q_i^t \cdot p_u. \tag{7}$$

The learning process is carried out by minimizing the function of the equation 8:

$$\min_{q^*,p^*,b^*} \sum_{(u,i)\in K} (r_{ui} - \mu - b_i - b_u - q_i^t \cdot p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2). \tag{8}$$

3 Experimental Work

In this section, two groups of data-driven experiments are reported, to observe the behavior of the studied models with respect to data with different noise levels: Behavior with respect to noisy data, and effect of LFs in the Regularized-SVD model. The implementation was done in Python, using Google's TensorFlow library.

Table 3. RMSE for each experiment.

Exp		Funk-SVD				Regularized-SVD				Bias-SVD			
		DS1		DS2		DS1		DS2		DS1		DS2	
	LFs	Min	Epoch	Min	Epoch	Min	Epoch	Min	Epoch	Min	Epoch	Min	Epoch
A	10	1.229	527	1.091	519	1.227	639	1.093	605	**0.778**	10,772	**0.946**	13,160
	20	1.764	581	1.393	511	1.568	2,818	1.338	617	**0.778**	11,321	**0.946**	12,360
	30	2.525	661	1.776	540	1.634	5,982	1.685	860	**0.779**	10,112	**0.945**	12,519
B	10	1.485	401	1.331	407	1.487	460	1.324	433	**0.972**	12,432	**1.122**	11,754
	20	2.126	441	1.672	424	1.794	770	1.642	466	**0.963**	12,345	**1.122**	12,832
	30	2.818	547	2.072	453	2.286	4,602	2.001	489	**0.970**	10,976	**1.122**	12,471
C	10	1.961	292	1.696	314	1.928	285	1.691	320	**1.375**	12,003	**1.436**	12,570
	20	2.600	289	2.001	306	2.489	341	1.979	323	**1.376**	10,564	**1.436**	12,806
	30	3.259	312	2.439	343	3.101	3,876	2.392	368	**1.370**	11,019	**1.436**	11,830
D	10	1.607	381	1.499	415	1.619	455	1.382	427	**0.968**	10,010	**1.122**	9,407
	20	2.415	472	1.797	413	2.228	3,121	1.787	438	**0.965**	11,478	**1.122**	11,798
	30	3.272	480	2.332	440	2.150	5,751	2.255	500	**0.965**	11,777	**1.123**	9,608
E	10	2.095	264	1.747	303	2.062	316	1.750	320	**1.380**	10,972	**1.441**	9,040
	20	2.882	309	2.186	317	2.793	353	2.154	300	**1.379**	11,543	**1.442**	10,325
	30	3.680	280	2.724	320	2.919	5,912	2.715	342	**1.379**	11,989	**1.441**	9,842
F	10	3.041	249	1.810	292	3.049	5,000	1.776	287	**1.547**	9,927	**1.440**	8,080
	20	4.558	65	2.448	296	2.107	5,878	2.424	280	**1.547**	7,847	**1.440**	10,676
	30	5.205	67	3.306	270	2.020	5,110	2.744	9,653	**1.548**	9,324	**1.440**	9,723

3.1 Behavior with Respect to Noisy Data

In this research, three ML models were implemented: Funk-SVD, Regularized-SVD, and SVD with biases or Bias-SVD. The results of training and testing with different degrees of noise were recorded, based on 10, 20 and 30 LFs. 25% of the data, taken at random, were considered for testing and 75% for training. The learning degree was set at $l_r = 0.01$ and the regularization constant $\lambda = 0.05$. The loss function uses MSE and Adam's algorithm, which is a faster variant of classical SGD. In each experiment, the minimum and the time at which it was reached before deregulation were recorded. Some error convergence curves are presented in Fig. 3, for each curve a window of the behavior of the models around the minimum is shown.

3.2 Data Used

Two MovieLens DataSets were used, with different distributions; some characteristics are shown in Table 1. Table 2 describes the variants that were made to each DataSet in Table 1, for experimentation. The original DataSets can be found in [9], and their random variants, as well as the complementary error convergence curves, can be found in [6].

4 Analysis of Results

Table 3 shows the results with respect to the test data, for both data sets (DS1 and DS2) and their variants. Fig. 3 shows the error convergence curves for DS1, the curves obtained for DS2 can be consulted in [6].

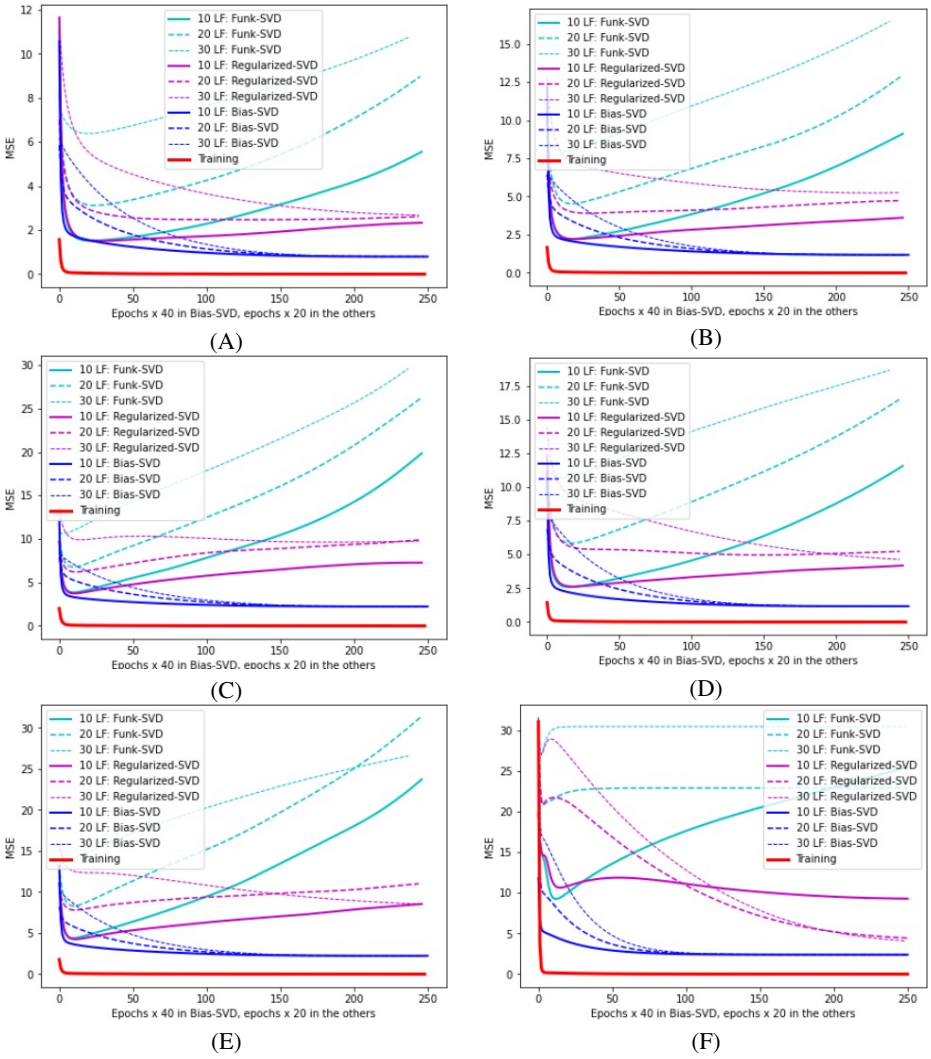


Fig. 3. Convergence curves of the experiments (A, B, C, D, E and F) in DS1.

The columns titled Min and Epoch show the minima and the epochs in which they were found. The results in bold ones are the best in each experiment. As can be seen in Table 3, the Bias-SVD algorithm was more accurate in all cases, while Funk-SVD had the worst performance. This is because Funk-SVD is not regularized.

In the curves of each experiment it can be seen that Funk-SVD converges faster towards the minimum in all cases, but soon deregulates, while Bias-SVD converges more slowly. In the Min columns for DS1 and DS2 of Bias-SVD the results with 10, 20 and 30 LFs are, in most cases, the same or very similar, while the corresponding results in the other two models increase as increasing the number of LFs. This indicates that the Funk-SVD and Regularized-SVD models require higher complexity than Bias-SVD.

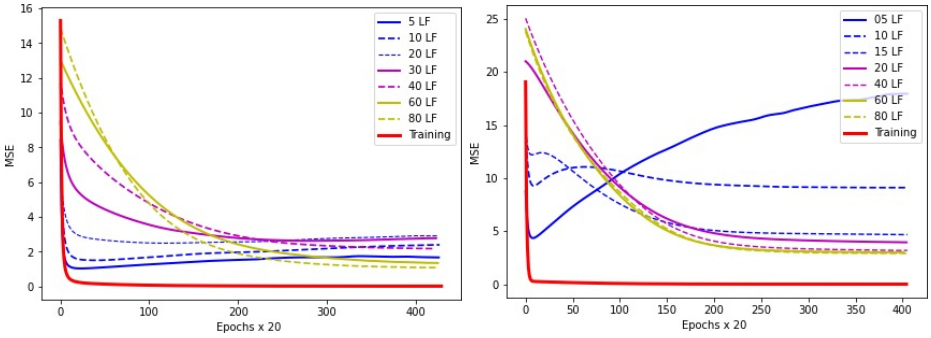


Fig. 4. MSE in the model Regularized-SVD and the experiments: A y F.

The results show that, despite the noise, Bias-SVD is not only more accurate but also more stable, although as can be seen as the noise increases the accuracy decreases, and in the worst case (exp. F) it even tends to become a little deregulated. In the curves of the experiments (A, B and C) for DS1, the same pattern of convergence is observed, this is because in experiment A there are no changes in the data, while in B and C random changes were made in relation to the position of the ratings and/or in relation to the frequency of each rating among the same movie rated.

In the curves of the experiments (D, E and F) for DS1, in Regularized-SVD, it can be observed that as the number of LFs increases, the convergence tends to improve (after 30 LFs in D and E), this is more noticeable in experiment F after 20 LFs, where there is a trend towards the Bias-SVD minimum, which seems to indicate that when the data is too noisy, the mechanisms that Bias-SVD-based algorithms have to deal with biases lead to slower but also more accurate convergence.

The best results for Bias-SVD were obtained in experiment A, both for DS1 and DS2, which was expected since they are the original data without changes, but it is striking that the results were not so bad in experiments B and D, with an error close to 1 for DS1, and close to 1.1 for DS2, This means that a score of 5 can be predicted as 4, that a score of 3 can be predicted as 2 or 4, which would not be far from reality.

4.1 Effect of LFs in the Regularized-SVD Model

In order to observe the behavior of the Regularized-SVD model in relation to the number of LFs and noisy data, the algorithm was run with 5, 10, 15, 20, 25, 30, 35, 40, 60 and 80 LFs. $10,000$ training epochs were implemented with $l_r = 0.01$ and $\lambda = 0.05$. Some convergence curves for experiments A and F are shown in Fig. 4.

The idea was to establish, through data-driven experiments, that increasing noise leads to the loss or weakening of hidden relationships that exist in the data. ML models learn these hidden relationships to make generalizations about unseen data, but if the data are noisy as in D, or completely chaotic as in F, one would expect the model to be unable to learn such relationships and therefore not could be generalized.

Table 4. RMSE in Regularized-SVD in each experiment with DS1 and DS2.

DS1									
	Experiment A			**Experiment D**			**Experiment F**		
LFs	**Training**	**Test**	**Minimum / Epoch**	**Training**	**Test**	**Minimum / Epoch**	**Training**	**Test**	**Minimum / Epoch**
5	0.600	1.30	1.015 / 560	0.767	1.817	1.274 / 388	0.713	**4.2**	2.087 / 263
10	0.448	1.549	1.226 / 873	0.549	2.154	1.618 / 432	0.157	**3.007**	3.007 / 10,000
15	0.345	1.722	1.423 / 1,625	0.377	2.359	1.968 / 533	0.138	**2.146**	**2.146** / 10,000
20	0.262	1.722	1.577 / 2,487	0.244	2.448	2.210 / 1,554	0.134	**1.973**	**1.972** / 9,954
25	0.202	**1.677**	1.613 / 4,544	0.16	**2.367**	**2.206** / 5,069	0.133	**1.88**	**1.88** / 10,000
30	0.159	**1.668**	1.619 / 5,726	0.137	**2.056**	**2.056** / 10,000	0.132	**1.842**	**1.842** / 10,000
35	0.132	**1.564**	**1.558** / 9,019	0.131	**1.789**	**1.789** / 10,000	0.131	**1.804**	**1.803** / 9,929
40	0.118	**1.473**	**1.469** / 9,910	0.128	**1.632**	**1.632** / 10,000	0.131	**1.78**	**1.78** / 10,000
60	0.111	**1.16**	**1.159** / 9,788	0.126	**1.387**	**1.387** / 10,000	0.13	**1.73**	**1.73** / 10,000
80	0.108	**1.037**	**1.037** / 10,000	0.124	**1.309**	**1.309** / 10,000	0.13	**1.705**	**1.705** / 10,000
DS2									
	Experiment A			**Experiment D**			**Experiment F**		
LFs	**Training**	**Test**	**Minimum / Epoch**	**Training**	**Test**	**Minimum / Epoch**	**Training**	**Test**	**Minimum / Epoch**
5	0.743	0.99	0.98 / 781	0.937	1.520	1.242 /439	1.173	2.033	1.590 / 307
10	0.618	1.500	1.102 /593	0.772	1.954	1.391/442	0.903	2.630	1.806/ 289
15	0.524	1.694	1.203/ 567	0.620	2.162	1.558 / 415	0.634	3.338	2.100 / 270
20	0.428	1.909	1.352 / 658	0.495	2.479	1.801 / 431	0.380	3.922	2.394 / 283
25	0.352	1.955	1.516 / 674	0.359	2.674	2.017 / 471	0.198	**3.882**	2.816 / 307
30	0.295	2.034	1.689 / 729	0.251	2.795	2.323 / 511	0.135	**2.760**	**2.759** / 10,000
35	0.229	2.012	1.827 / 3,360	0.168	**2.792**	2.568 / 577	0.122	**2.271**	**2.271** / 10,000
40	0.180	2.027	1.899 / 4,769	0.127	**2.493**	**2.482** / 9,984	0.126	**2.041**	**2.061** / 10,000
60	0.094	**1.629**	**1.629** / 10,000	0.104	**1.647**	**1.647** / 10,000	0.109	**1.763**	**1.762** / 9,979
80	0.086	**1.258**	**1.258** / 10,000	0.101	**1.441**	**1.441** / 10,000	0.107	**1.665**	**1.665** / 10,000

5 Analysis of Results

Table 4 shows the results of the Regularized-SVD model studied, for the training and test data of the experiments for DS1 and DS2, and its variants: A, D and F, and Fig. 4 shows the convergence curves for MSE. Note that, in experiment A for DS1, by increasing from 5 to 20 LFs the error increases in the test data, the minimum also gradually increased from 5 to 30 LFs and these were found, each time at more distant times. For DS2 the minimum grows until it reaches 40 LFs, then the decrease begins.

These increments of the minimum mean that the model is not learning the hidden relationships between the training and testing data. In the curves of experiment A in Fig. 4 it can be seen that the model begins to learn from 30 LFs (see Table 4, in minimum/epoch column), and its best performance is reached in 80 LFs. Thus, as the model begins to learn, the error decreases as the number of LFs increases.

On the other hand, in experiment D for DS1, the results show that the error begins to be minimized from 25 LFs onwards, although it cannot yet be generalized, because the error of the test data is still large, however, there are indications that the model learns more difficult in experiment A than in D. A similar pattern can be observed in DS2, with the difference that convergence is slower. Finally, in experiment F for DS1 (Fig. 4), the DataSet used is too noisy, and as can be seen in Table 4, the final error of the

test data is always lower as the number of LFs increases, and the minimum begins to decrease from 15 LFs (for DS2 the decrease of the minimum begins after 25 LFs), it could be said that from there the model begins to learn, since the training error after 10 LFs (after 20 LFs for DS2) could be considered small. Therefore, it appears that the learning process of the studied algorithm requires fewer LFs with noisy data (F) than with data without noise (A). One explanation is that, in experiment A, there are many hidden relationships in the original data, so a larger number of LFs are needed so that the model is not so simple.

If the model is intended to learn with few LFs, the underfifting problem is generated, therefore, the ML model for experiment A needs a little more complexity. While in the chaotic data of experiment F, the hidden relations have been weakened or lost, so the model gives the impression of learning with fewer LFs than in experiment A. However, it could be interpreted that, as the hidden relationships are weakened or lost, the model does not learn but memorizes the configuration of the data, so the model for F may be simpler, although less precise than in experiment A.

6 Conclusions

In this work, two groups of experiments have been presented to observe the behavior of classical SVD models with respect to noisy data. Two MovieLens DataSets with different distributions were used. For each DataSet, 5 variants were configured in which a certain level of randomness is introduced into the data. The error was measured using RMSE and Adam's algorithm was used as the optimizer.

In the first group of experiments, 10, 20 and 30 LFs were used. In experiment A, the smallest error in the test data was $\mathrm{RMSE} = 0.778$ for DS1 and $\mathrm{RMSE} = 0.945$ for DS2 in the Bias-SVD model, while in the noisier DataSets the smallest error was 1.547 for 10 and 20 FLs in DS1, and 1,440 for DS2, also in the Bias-SVD model. It is concluded that despite the noise, Bias-SVD performs better than the others. In the second group of experiments, the Regularized-SVD model was tested with 5 to 80 LFs, to observe the effect that LFs have on noisy data. In these experiments it was found that as the noise in the data increases, the algorithms need a smaller number of LFs to initiate error convergence.

This can be interpreted in the sense that in distorted or completely chaotic data, the hidden relationships between the training and test data are weakened or have been lost, so the algorithms do not need to be very complex to learn. As future work, we will continue experimenting with noisy data, to try to identify patterns that can help us address potential sabotage that may exist in RS.

Acknowledgments. Supported by Universidad Tecnológica de Izúcar de Matamoros.

References

1. Ahuja, R., Solanki, A., Nayyar, A.: Movie recommender system using k-means clustering and k-nearest neighbor. In: 9th International Conference on Cloud Computing, Data Science and Engineering, pp. 263–268 (2019) doi: 10.1109/confluence.2019.8776969

2. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowledge-Based Systems, vol. 46, pp. 109–132 (2013) doi: 10.1016/j.knosys.2013.03.012
3. Brunton, S. L., Kutz, J. N.: Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press (2019) doi: 10.1017/9781108380690
4. Chipman, J. S.: Multicollinearity and reduced-ranked estimation. Lectures in Econometric Theory, pp. 60–81. Minneapolis: University of Minnesota
5. Compton, E. A., Ernstberger, S. L.: Singular value decomposition: Applications to image processing. Citations Journal of Undergraduate Research, Lagrange College, vol. 17, pp. 99–105 (2020)
6. Drive (2023) drive.google.com/drive/folders/1j2r1facWa0tmHucUGwjwJrONaKjjxVSO?usp=sharing
7. Eckart, C., Young, G.: The approximation of one matrix by another of lower rank. Psychometrika, vol. 1, no. 3, pp. 211–218 (1936) doi: 10.1007/bf02288367
8. Gouvert, O., Oberlin, T., Févotte, C.: Negative binomial matrix factorization for recommender systems (2018) doi: 10.48550/ARXIV.1801.01708
9. GroupLens (2023) grouplens.org/datasets/movielens/
10. Huang, C., Wang, X., He, X., Yin, D.: Self-supervised learning for recommender system. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (2022) doi: 10.1145/3477495.3532684
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer, vol. 42, no. 8, pp. 30–37 (2009) doi: 10.1109/mc.2009.263
12. Kuandykov, A. A., Rakhmetulayeva, S. B., Baiburin, Y. M., Nugumanova, A. B.: Usage of singular value decomposition matrix for search latent semantic structures in natural language texts. In: 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (2015) doi: 10.1109/sice.2015.7285567
13. Kuraparthi, S., Kollati, M., Kora, P.: Robust optimized discrete wavelet transform-singular value decomposition based video watermarking. Traitement du Signal, vol. 36, no. 6, pp. 565–573 (2019) doi: 10.18280/ts.360612
14. Lucero-Alvarez, C., Quintero-Flores, P. M., Ortiz-Ramirez, C. A., Mendoza-Crisostomo, P., Montiel-Hernandez, J., Vazquez-Vazquez, M.: Didactic evaluation of some useful predictive methods in neighborhood-based recommendation systems. In: IEEE Mexican International Conference on Computer Science, pp. 1–8 (2022) doi: 10.1109/enc56672.2022.9882912
15. Lucero-Alvarez, C., Quintero-Flores, P. M., Perez-Cruz, P., Ortiz-Ramirez, C. A., Mendoza-Crisostomo, P., Montiel-Hernandez, J.: Literature review on information filtering methods in recommendation systems. In: Mexican International Conference on Computer Science, pp. 1–8 (2021) doi: 10.1109/enc53357.2021.9534807
16. Mamani-Roque, M.: Análisis semántico latente mediante descomposición en valores singulares (2018)
17. Ramírez-Morales, C. A.: Algoritmo SVD aplicado a los sistemas de recomendación en el comercio. Tecnología Investigación y Academia, vol. 6, no. 1, pp. 18–27 (2018)
18. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Application of dimensionality reduction in recommender system-a case study. University of Minnesota Minneapolis, Deparment of Computer Science and Engineering (2000)
19. Simon Funk Homepage (2023) sifter.org/~simon/journal/20061211.html
20. Su, J., Guan, Y., Li, Y., Chen, W., Lv, H., Yan, Y.: Do recommender systems function in the health domain: A system review (2020) doi: 10.48550/ARXIV.2007.13058
21. Wang, H.: Dotmat: Solving cold-start problem and alleviating sparsity problem for recommender systems. pp. 1323–1326 (2022) doi: 10.48550/ARXIV.2206.00151

22. Wang, H.: Zeromat: solving cold-start problem of recommender system with no input data. In: IEEE 4th International Conference on Information Systems and Computer Aided Education, pp. 102–105 (2021) doi: 10.1109/iciscae52414.2021.9590668
23. Witten, D. M., Tibshirani, R., Hastie, T.: A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics, vol. 10, no. 3, pp. 515–534 (2009) doi: 10.1093/biostatistics/kxp008

ISSN 1870-4069

Characterization of Honey Bee Anatomy for Recognition and Analysis Using Image Regions

Apolinar Velarde-Martínez, José Daniel Reyes-Moreira,
Juan Carlos Estrada-Cabral, Gilberto Gonzalez-Rodriguez

Instituto Tecnológico El Llano Aguascalientes,
Ingeniería en Agronomía,
Mexico

{apolinar.vm, daniel.rm, juan.ec,
gilberto.gr}@llano.tecnm.mx

Abstract. Extraction of object features in an image, for automatic recognition, is one of the main parts of a computer vision project. Extracting features of the objects in an image from a real environment, without controlling factors that influence the acquisition of the image, has a high degree of complexity. This paper describes a method to characterize the anatomy of honey bees, with images acquired at the entrance of the hive in an apiary using digital image processing and automatic classification methods for recognition and analysis. The method was tested with a base of 1050 real test images divided into incremental subsets of 50 images. Recognition was performed with the Support Vector Machine (SVM) and k-Nearest Neighbors (kNN). Two performance parameters were evaluated, the percentage of correctly classified images with the presence of the bee in the entrance of the hive and the classification time for each evaluated subset. The results of the experiments show better recognition percentages with the SVM while the recognition times of the image subsets using the kNN method are better.

Keywords: Apiary, honey bee, automatic recognition, support vector machine, k-nearest neighbors.

1 Introduction

Object detection is a fundamental visual recognition problem in computer vision [18]. When this problem is addressed with image objects in an environment of controlled factors and in a specific position, the implementation of the techniques to recognize the objects is not complex. On the other hand, when this problem is addressed with the recognition of objects contained in images of a real environment and in different positions of the object, the degree of complexity when implementing techniques for feature extraction and recognition increases.

In a real environment, different factors influence the images that are acquired. The first factor to consider is the lighting that is related to the environmental conditions, i.e., if the images are acquired during the day or at night, on a cloudy day, with rain, with a lot of wind or with full sun, the lighting affects the surface and the objects to be analyzed.

A second factor is the area of the real environment from which the images are extracted for the analysis of the objects; it must be considered if the objects are in movement or they are static. Consider whether the objects being analyzed are inanimate or living beings; inanimate objects in motion are generally moved by a device to regulate their speed; in case of living beings, the complexity is greater, since the movement is unpredictable, their positions cannot be controlled.

Then, when the objects present movement and different positions in the image, more distortions must be treated in the processing with more complexity and higher processing speeds [18]. Due to the above, it is necessary to apply digital image processing techniques and object recognition in the image with different strategies for detection and recognition objects in the scene; many strategies have been proposed in the literature.

This research work deals with the recognition of objects in different environmental conditions, in a real environment and with living beings. The strategy of using regions of interest in the image has been chosen [18, 17] a technique similar to two techniques called block-oriented image decomposition structures [11] and the technique based on the subdivision of the image matrix into four quadrants of equal size [4].

With the above described, this paper presents and describes an investigation for the automatic recognition of bees in the entrance of the hive. The recognition is made from images taken from an apiary with live bees that have different positions. This research is the continuation of the work presented in [15], and its main objective is to develop a method with the three most common phases of automatic shape recognition, namely, digital image processing, the extraction of the characteristics (namely, characterization) of objects in the images and the recognition of the object in the image.

Digital image processing is carried out with different techniques as explained in the following sections; extraction of characteristics of objects is made using the coloring method [13, 1] and the Freeman Chain Code method [14, 5, 2, 8] for the generation of feature vectors [11, 2, 8]; and the recognition of the object in the image is carried out with the Support Vector Machine (SVM) and k-nearest neighbors (kNN) [17, 3] methods. Each of the three phases that constitute this research project is explained in the following sections of this work together with the techniques used in each phase.

In this work we use the terms honey bee and object of the image interchangeably to refer to the presence of the honey bee at the entrance of the hive. In section 2, some works related to this research are described. Section 3, problem statement, describes the general environment of this research. The basic concepts used are described in section 4, for a better understanding of the problem. Section 5 phases of the project, explains the way in which each of the three phases that constitute the proposed method was implemented; Section 6 presents the results obtained with the experiments carried out. Finally, section 7 presents the conclusions of this research project.

2 Related Works

Due to the fact that this research has several research areas, due to space issues, only a very small number of works related to the recognition of objects by identifying the regions of the images are commented in this section.

The concept of regions in image processing has been extensively studied in different research papers [17, 4, 13, 1]. The seminal work described in [4] makes use of quad-trees to represent image regions and obtain a simple type of boundary representation.

In [1, 9], the use of a reduced number of labels, which does not exceed a certain amount, is proposed for the identification of the regions, and ensure that two neighboring regions with the same label cannot exist; information about some region pixel is added to the description so that this can provide a complete region reference.

All information is stored in a separate data structure. A widely used algorithm is proposed in [12], and applies to images encoded by sequence length and on images represented as straightforward matrices. Other technique, proposes a model to address the classification problem by detecting if a region contains both "background" and "foreground" regions [17].

Moreover, in [11] a block-oriented image decomposition structure can be used to represent image content in image database system. In [1] image regions are used to classify only a specific region of the image that corresponds to a given object using Convolutional Neural Nets (CNN). Considering the works described in this section, in this research the regions of the images are used as areas of interest to extract characteristics of the object and allow the recognition of the object in the image.

3 Problem Statement

Application of technologies for the care and preservation of species in the world is necessary. Sometimes, actions of human carry out to stop the disappearance and loss of animal species are not enough, therefore, technological developments are needed to help in the early detection of diseases or parasite attacks on species.

This research work has been divided into three stages. The first stage was published in [15], the second stage is presented in this paper, and the third stage is currently being worked on.

So, this second phase of the research project focuses on the detection of the presence of bees at the entrance of the hive with the use of images of the real environments in the apiaries, which will lead us in a future work to the detection of ectoparasites in honey bees. Some reasons why this research is justified are the following:

- Develops an automated system for the detection of bees in the entrances of the hives, using as a base the communication platform installed between an apiary and a cluster of servers [15].
- Explore the research area of object recognition in real settings, with living beings, using the technique of detecting objects in image regions.
- This work will serve as the basis for the current development of an automated visual inspection system in real time, to detect ectoparasites in honey bees and contribute to the care and preservation of honey bees in the "Region del Llano", which is located between the states of Jalisco and Aguascalientes, Mexico. In this region, honey bee is cultivated for honey production and also as a pollinator of crops.

4 Basic Concepts

For a better understanding of this research, this section defines a set of terms that will be used in the following sections:

– Image. An image is a spatial representation of an object, a two-dimensional or three-dimensional scene [6]; this can be modeled by a continuous function of two variables $f(x,\ y)$ where $(x,\ y)$ are coordinates in a plane [13].

– Region. A region is a connected subset of a $2^n - by - 2n$ array, which is made up of unit-square "pixels" [4].

– Segmented Image. From [13] we obtain that a segmented image R consists of m distinct, disjoint regions R_i, as clearly shown in equation 1. The image R consists of objects and a background:

$$R_b^c = \bigcup_{I=1, i \neq b}^{m} R_i, \tag{1}$$

where R^c is fixed complement, R_b is considered background, and other regions are considered objects.

5 Project Phases

This section describes the way in which each of the three phases of the research project was implemented. Digital image processing phase shows the techniques applied in each step of image processing; the characterization phase of the regions in the image, mentions the algorithms applied to label each region of the image, the features calculated for each region, and the way in which the feature vectors are built; finally, the recognition of the object in the image, mentions the two techniques applied to recognize the presence or absence of the honey bee in the image; for better under- standing of the reader, the techniques applied in each phase are mentioned.

5.1 Digital Image Processing

In this phase, the digital treatment of the images extracted from the apiary is used; the processes applied are image cropping [19], the conversion from a RGB (Red, Green and Blue) image to gray [12], Gaussian blur, The Canny edge detector [14, 10] and contour detection [14].

For the example of the digital treatment of the image, consider image 1, this image (like all the images that are processed) is extracted from the apiary, and acquired in the entrance of the hive [15]; a white background is adequate on the surface of the entrance to highlight the presence of bees; figure 1 shows the presence of 6 live bees; 4 bees are considered as complete objects, and 2 bees segmented by the acquisition equipment; of the two segmented bees, one bee shows only the abdomen and the head of the second bee appears with a frontal shot in the image.

Fig. 1. An image taken from the apiary and acquired in the entrance of the beehive.

Please let me comment, this image has been selected from set of images, because it is suitable to explain each step of the digital treatment of images. Each row of image 2 shows the results of applying the 4 processes to the original image 1. In the following sections, each of the processes is explained in a reduced way, due to space issues:

– **Image cropping:** Due to the actual size of the extracted image, a cropping process [19] is necessary. This process allows to divide the image and obtain three smaller images that allow to improve the processing. The three images in row 2b represent the image after the cropping process.

– **Conversion from RGB image to gray:** The process of converting the image from RGB to grayscale is to simplify the algorithms and also remove complexities related to computational requirements. Image 2c shows each image after the RGB image to gray conversion process.

– **Gaussian Blur:** A necessary function to eliminate mute noise is necessary after the conversion to grayscale. Then a Gaussian blur for noise is applied. The image 2d shows each image after the process of making the conversion to Gaussian blur.

– **Canny edge detector:** An Operator for edge detection is applied after Gaussian blur. The Canny edge detector [14, 10] was selected, like a multi-stage algorithm to detect the wide range of edges in the image. Image 2e shows each image after the process of applying the Canny edge detector.

– **Contour detection:** Contours detection is an important image processing technique [14], and is a process for curve joining all the continuous points (along with the boundary), having same color or intensity. In this research work we are evaluating the application of some techniques proposed in the literature. Obtained results are explained in the next sections.

5.2 Characterization of Regions in the Image

After the digital process of image, identification of regions in the image is started. The identification of regions in the image is the most important process in this work, because the identification of object shape allows us the complete recognition of the bee.

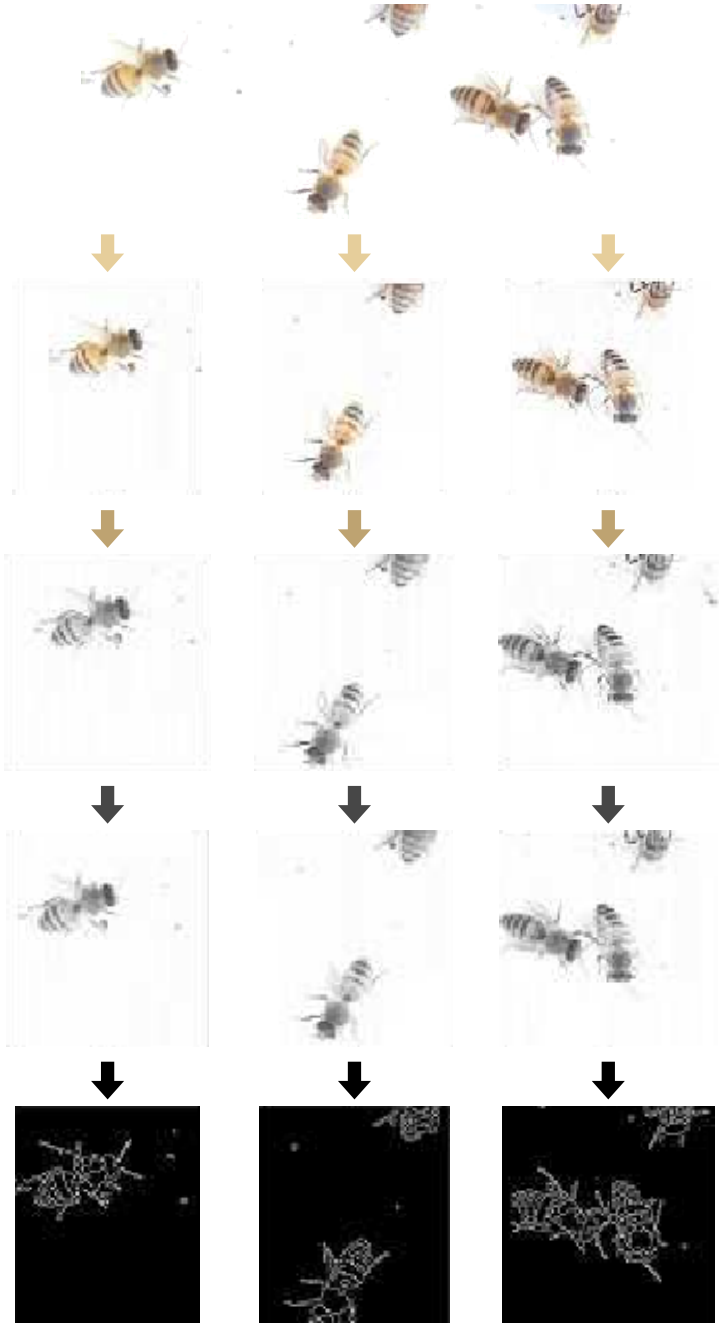
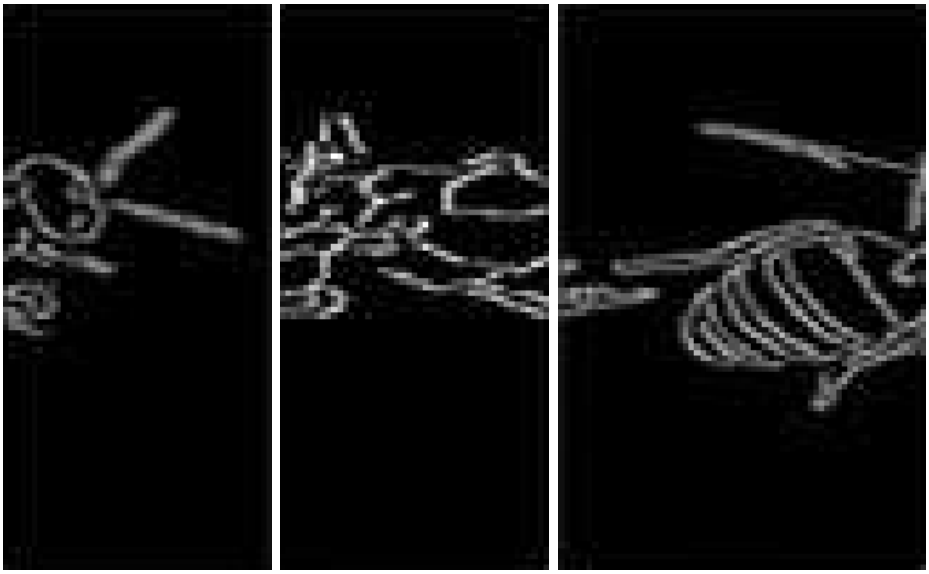


Fig. 2. Original image to which each of the phases of digital image processing is applied. The Image in row 1a is the original image, the images in rows 1b, 1c, 1d and 1e show the original image after the cropping, RGB image to gray, Gaussian Blur and Canny Edge detector processes respectively.



(a) (b) (c)

Fig. 3. Segmentation of the image of the hive into its constituent parts, for the detection of regions; a) head and antenna, b) thorax, c) abdomen and hind legs.

This identification is carried out by segmenting the images into different semantically significant regions [12, 16], to detect the objects and edges in the image [7]. Given the above conditions, to recognize the parts of the bee, this paper proposes the method of labeling or coloring (also called, connected component labeling) [13, 1, 12], to label each region with a unique integer.

In this work, the input to the labeling algorithm are binary images (with images produced after contour detection), where the background is represented by pixels with zero value and the objects by non-zero-pixel values. The result after labeling is an image with the background represented with zero values and the regions represented with non-zero labels. Each region is characterized and this information is stored in a separate data structure called the feature vectors [11] of the regions.

In the next phase for the recognition of the object in the image, mathematical morphology approaches are used for region identification. In the following paragraphs, the way in which the isolation of the regions in the image is carried out and the generation of the feature vectors [11, 8, 9] of each region found is explained, as well as the discrimination noise in the image for object identification.

Please consider that the images obtained with the contour detection algorithm (the three images in row 2e of image 2) are not cropped; for explanation purposes 1 of row 2e is cropped as shown in image 3.

Consider images come from a real environment, this causes the definition of the contours in the last phase of the image processing (consider the first image of the row 2e) of image 2), are not defined correctly and the shape of the bee is not complete for recognition. Image 3, shows the first image of row 2e of figure 2 in a cropped form.

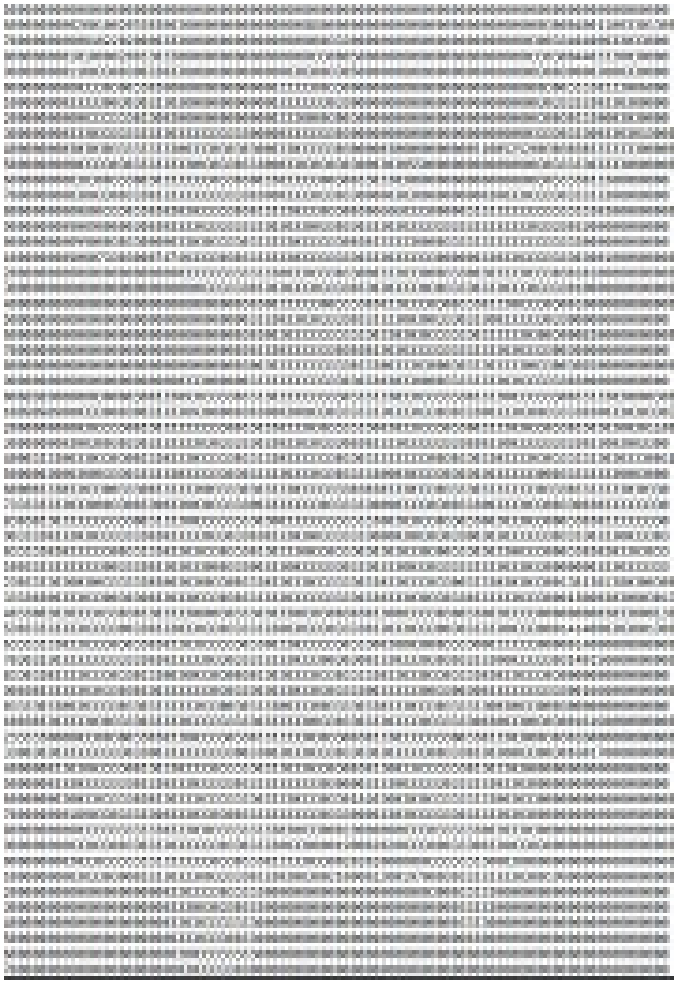


Fig. 4. Result of applying the coloring algorithm to the first image of row 2e of image 2.

The division into different semantically significant regions lead the example presented here; it is clear that the contours of the head and the antennae of the bee are defined, but not united, which produces that the shape of this part of the bee is partially complete, see image 3a. One way to deal with the problem of incomplete contours or noisy contours is to apply thinning techniques to the image, but in our case, these techniques were not suitable for the expected results.

For the next two parts of the bee, the thorax and the abdomen, Figures 3b and 3c respectively, the contours are not joined; these contours are segmented into several parts and incomplete, even the curvilinear shape of the thorax is not shown with contours. Despite this, some parts of the bee can be recognized, for example, the abdomen and hind legs can be recognized if the algorithm is able to solve the segment joining problem.

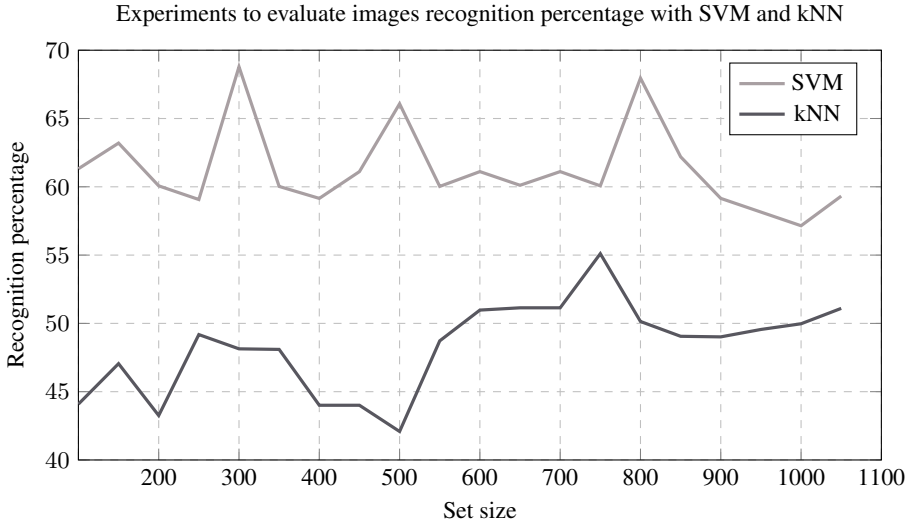


Fig. 5. Recognition percentages for each set of test images, applying SVM and kNN techniques.

When encountering this problem, the strategy used in this work is to identify the non-joined contours through the regions; and then characterize each region to finally perform a calculation of closeness between regions. If the regions identified as antennae are close to a region identified as the head, then we are identifying the frontal part of the bee. Obviously, the classifiers used previously "know" the regions of the bee.

Considering the segmented contours in the image, the coloring algorithm was used directly after edge detector process. In the next section the process of applying the Coloring algorithm, Freeman Chain Code algorithm and discrimination of noise in the image are explained.

Application of the Coloring algorithm. The result of applying the coloring algorithm [13, 1], can be seen in image 4. Each of the regions are segmented with identifiers after the execution of the algorithm. We can observe the following: Near regions can have near region numbers. That is, if we consider an antenna of the bee, it will have a number of close regions of the head or of the other antenna, which allows us to apply the Manhattan algorithm to find the closeness of regions.

Application of the Freeman Chain Code algorithm. This algorithm string records the movement of tracker during complete tracing of character structure, from which shape primitives, consists of simple line and curve shapes [8]. Once the regions have been identified in the image, the Freeman Chain Code algorithm [14] is executed to identify the features of each region.

This algorithm again scans the image for the extraction of the features of each region. The first characteristics calculated for each region are: the perimeter, the density and the area, which are stored in the vector of characteristics of the region in question. In a second cycle, the algorithm computes the corners, concavities, and moments of each region; in the same way the results of the calculations are stored in the feature vectors.

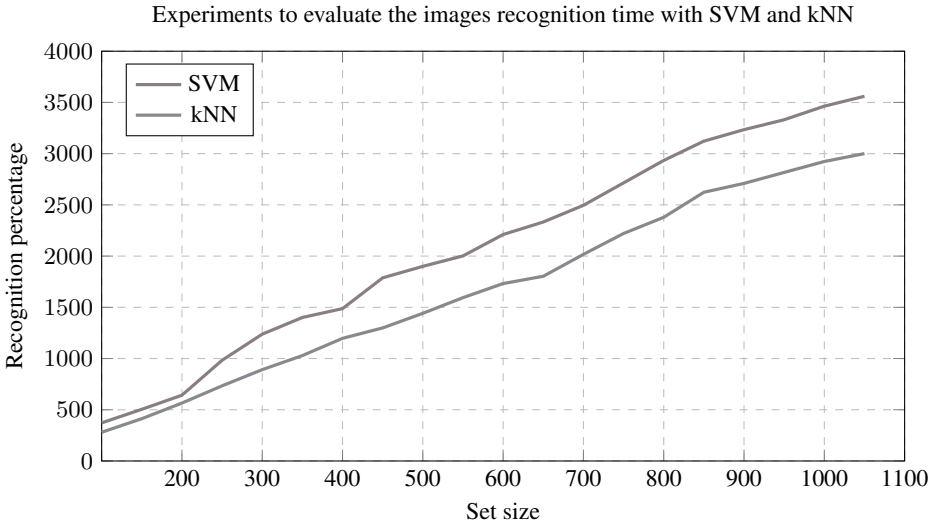


Fig. 6. Image processing time, applying SVM and kNN techniques.

Noise discrimination in the image. The noise in an image are regions or parts of the image that must be discriminated to achieve a better recognition of the object. Removing noise from the image is a process that is difficult to develop, but necessary because if it is not eliminated, the object is not recognized, it is partially recognized, or it causes confusion of the object with parts of the image that do not correspond to the object. In this work, the noise discrimination or elimination process occurs when the characteristics of the regions not recognized as part of the bee are extracted, and do not match with any of the parts of the bee.

Otherwise, if the algorithm indicates, this region is coincident with some part of the bee, other parts of the bee must be close or close in the neighborhood of regions. To calculate the closeness between regions, the distance from Manhattan is used. As a condition to enforce the member- ship of a region to the bee, it must be true that a region has a distance from Manhattan relative to at least 3 neighboring regions. Obviously, it is a condition that is not verified for the first three regions found in the image.

5.3 Recognition of the Object in the Image

In a summary form due to space, this section explains the third phase, which is the recognition of the object in the image using the SVM and kNN methods. This phase of the project proceeds as follows, the generated feature vectors serve as input to the two methods to allow the identification of each of the parts of the honey bee such as antennae, head, thorax, abdomen and limbs (paws).

By joining the recognitions of each part of the bee represented by the feature vectors, it is determined whether the object in the image is the honey bee or not the honey bee in the image. The lack of feature vectors of some parts of the bee is decisive to discriminate the image and determine if the object (honey bee) does not exist.

6 Experiments

For the experiments explained here, a base of 1050 images were processed without discrimination, i.e., the images were acquired over a period of one month, once the remote image acquisition system of the apiary was put into operation. To carry out the experiments, the image base was divided into incremental groups of 50 images. Two parameters were evaluated, the recognition percentage and the image processing time. The recognition percentage allows knowing the number of images that were recognized in each group of images that is evaluated. This parameter also makes it possible to compare the results of the two recognition techniques and observe the effectiveness of each technique.

The processing time is important because one of the objectives of this research project is the processing of images in real time for the detection of ectoparasites in bees, so to determine which recognition technique is the fastest, this parameter was evaluated. Another important aspect to consider in the experiments was the position of the bees without rotation, i.e. the bees in the image are considered in the same position because in this investigation, the environmental conditions (illumination) that pre-vailed during the acquisition of the images in the apiary, were considered.

Also, in these experiments the six calculated characteristics were considered. In future work, we consider dividing the features into two subsets of three features each, with the aim of improving processing times and observing the incidence of each feature on object recognition. With the above described, the results obtained so far with each parameter evaluated are described in the following paragraphs.

6.1 Recognition Percentage

This experimentation was carried out to know the effectiveness of the SVM and kNN techniques together with the generated feature vectors. Graph 5 shows the recognition percentages for each set of test images. The graph shows the effectiveness of the SVM technique; it provides better results as a recognition technique using feature vectors. As it is possible to observe, as the number of images increases, the recognition percentage does not decrease compared to the kNN technique.

Here, we consider the number of images recognized by SVM can help the results in a real recognition environment. The kNN technique is optimal for smaller image sets, but as the number of images increases, classification is complicated. Although the recognition rates increase when the number of images in the set is 500 or more, it is not enough to outperform SVM.

6.2 Image Processing Time

Graph 6 shows the times obtained in the experiments with different sets of processed images. kNN stands out as a technique with shorter recognition time, when classify- ing the objects in the images; kNN is a technique with a very fast recognition convergence, which allowed to classify the images of each subset in less time compared to SVM; according to the increase of images in the sets of processed images, the response time of the classifier remains stable and is robust; although it should be noted, evaluation

tends to generate more false positives than the SVM classifier. SVM shows itself in the experiments as a more stable classifier with longer time, but with fewer false positives, i.e. it is more accurate in the classification.

7 Conclusions

This work describes a computer vision system, whose mission is to identify the presence or absence of bees in the entrance of the hive. The system performs in a real apiary environment and considers the most important factors affecting image acquisition. The way to achieve the classification is through the application of digital image processing techniques and techniques for the recognition of the object in the scene.

Two common classifiers in the literature were evaluated, with 2 performance parameters, the recognition percentage and the recognition time of the objects in the image with two supervised classifiers, kNN and SVM. The results obtained in experiments show SVM as a classifier with better responses, although more time consuming. kNN has better response times but with more errors when classifying the objects. These results are being considered in the next phase of this project.

Also, this work demonstrates the feasibility of using new technologies for the care and preservation of species. This work is constituted as a part of an integrative project whose final objective is the detection of ectoparasites in bees in real time. The final phase is currently being developed, which integrates the project proposed in [15], and this research work.

8 Future Works

Future research work is to develop the subsequent analysis of each of the parts of the honey bee, to detect the presence of attacks by ectoparasites and help the conservation and care of this species.

References

1. Appel, K., Haken, W.: Every planar map is four colorable. Part I: Discharging. Illinois Journal of Mathematics, vol. 21, no. 3 (1977) doi: 10.1215/ijm/1256049011
2. Azmi, A. N., Nasien, D.: Feature vector of binary image using freeman chain code (FCC) representation based on structural classifier. International Journal of Advances in Soft Computing and its Applications, vol. 6, no. 2 (2014)
3. Bzdok, D., Krzywinski, M., Altman, N.: Machine learning: Supervised methods. Nature Methods, vol. 15, no. 1, pp. 5–6 (2018) doi: 10.1038/nmeth.4551
4. Dyer, C. R., Azriel, R., Hanan, S.: Region representation: Boundary codes from quadtrees. Communications of the ACM, vol. 23, pp. 171–179 (1980)
5. Freeman, H.: On the encoding of arbitrary geometric configurations. IEEE Transactions on Electronic Computers, vol. EC-10, no. 2, pp. 260–268 (1961) doi: 10.1109/tec.1961.5219197
6. Haralick, R. M., Shapiro, L. G.: Glossary of computer vision terms. Pattern Recognition, vol. 24, no. 1, pp. 69–93 (1991) doi: 10.1016/0031-3203(91)90117-n

7. Linares, O. A., Botelho, G. M., Rodrigues, F. A., Neto, J. B.: Segmentation of large images based on super-pixels and community detection in graphs. IET Image Processing, vol. 11, no. 12, pp. 1219–1228 (2017) doi: 10.1049/iet-ipr.2016.0072
8. Nasien, D., Yulianti, D., Omar, F. S., Adiya, M. H., Desnelita, Y., Chandra, T.: New feature vector from freeman chain code for handwritten roman character recognition. In: 2nd International Conference on Electrical Engineering and Informatics, pp. 67–71 (2018) doi: 10.1109/icon-eei.2018.8784340
9. Nelson, R., Wilson, R. J.: Graph colourings. Longman Scientific and Technical, 1st ed (1990)
10. Rasche, C.: Rapid contour detection for image classification. IET Image Processing, vol. 12, no. 4, pp. 532–538 (2018) doi: 10.1049/iet-ipr.2017.1066
11. Remias, E., Sheikholeslami, G., Zhang, A.: Block-oriented image decomposition and retrieval in image database systems. In: Proceedings of International Workshop on Multimedia Database Management Systems, pp. 85–92 doi: 10.1109/mmdbms.1996.541858
12. Rosenfeld, A., Kak, A. C.: Digital picture processing. Computer Science and Applied Mathematics, vol. 1, 2nd ed (1982)
13. Sonka, M., Hlavac, V., Boyle, R.: Image processing, analysis, and machine vision. Cengage Learning, 1st ed (2015) doi: 10.1007/978-1-4899-3216-7
14. Vaddi, R., Boggavarapu, L. N. P., Vankayalapati, H. D., Anne, K. R.: Contour detection using freeman chain code and approximation methods for the real time object detection. Asian Journal of Computer Science and Information Technology, vol. 1, no. 1 (2013)
15. Velarde-Martínez, A., González-Rodríguez, G., Ibarra-Rodriguez, M. T., Orozco-Cortéz, B.: Video surveillance of beehives using computer vision and IoT. Research in Computer Science, vol. 151, no. 10 (2022)
16. Verdoja, F., Grangetto, M.: Efficient representation of segmentation contours using chain codes. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1462–1466 (2017) doi: 10.1109/icassp.2017.7952399
17. Wang, W., Zhang, A., Song, Y.: Identification of objects from image regions. In: Proceedings of the International Conference on Multimedia and Expo, pp. 1–253 (2003) doi: 10.1109/icme.2003.1220902
18. Wu, X., Sahoo, D., Hoi, S. C.: Recent advances in deep learning for object detection. Neurocomputing, vol. 396, pp. 39–64 (2020) doi: 10.1016/j.neucom.2020.01.085
19. Yan, J., Lin, S., Kang, S. B., Tang, X.: Learning the change for automatic image cropping. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 971–978 (2013) doi: 10.1109/cvpr.2013.130

ISSN 1870-4069

Tuning Control Law Gains in an Exoskeleton through Swarm Intelligence

Gerardo Adrián De La Rosa-Hernández, Griselda Quiroz-Compeán, Juan Angel Rodríguez-Liñan, Luis Martín Torres-Treviño

Universidad Autónoma de Nuevo León,
Facultad de Ingeniería Mecánica y Eléctrica,
México

gerardo.delarosahe@uanl.edu.mx

Abstract. Exoskeletons are mechatronic devices that can be used in physical rehabilitation programs for people who have suffered neurological injury such as ischemic or hemorrhagic brain accident. This type of injury can cause partial or total loss of mobility, so it becomes necessary to have assistive devices that support people to recover their mobility and return to their daily life. A typical problem in exoskeleton control is the adjustment of control law gains because it is challenging and lacks precision. For this reason, it seeks to know the appropriate values of the gains of a controller that generate the appropriate input torque in the exoskeleton's joints to solve the low-level control problem, managing to obtain the smallest possible error between the input reference path and the current angular position. To know the appropriate parameters, it is assumed that with the use of optimization tools such as evolutionary computing algorithms or collective intelligence the error signal in a multiple-joint structure, such as the exoskeleton, could be minimized. Comparison between heuristic and intelligent methods shows that objective function could be minimized to obtain best gains values, therefore, intelligent method calculates better performance values in terms of angular position error, so considering the calculated gains, the low-level control has been improved to accomplish with trajectory tracking problem.

Keywords: Exoskeleton, mechatronic assistive device, PID intelligent tuning, particle swarm optimization, integral squared error.

1 Introduction

As per the World Health Organization (WHO), more than 15% of people worldwide experiences some form of disability [5]. In the year 2010, in the United States exclusively, approximately 30.6 million individuals faced disability-related challenges with their mobility, such as having trouble in walking, climbing stairs or descending them, assistance of a wheeled chair, zimmer frame, cane or walking sticks [3].

Talking about the European Union in the year of 2011, is mentioned that there are over 11,000 people with walking disabilities [6]. Based on the National Health and Nutrition Survey of 2012, at a nationwide scale, approximately 4.9% of males and 5.8% of females (equivalent to around 2.5 million and 3 million individuals, respectively) were documented to have a disability affecting their mobility or ability to walk [8].

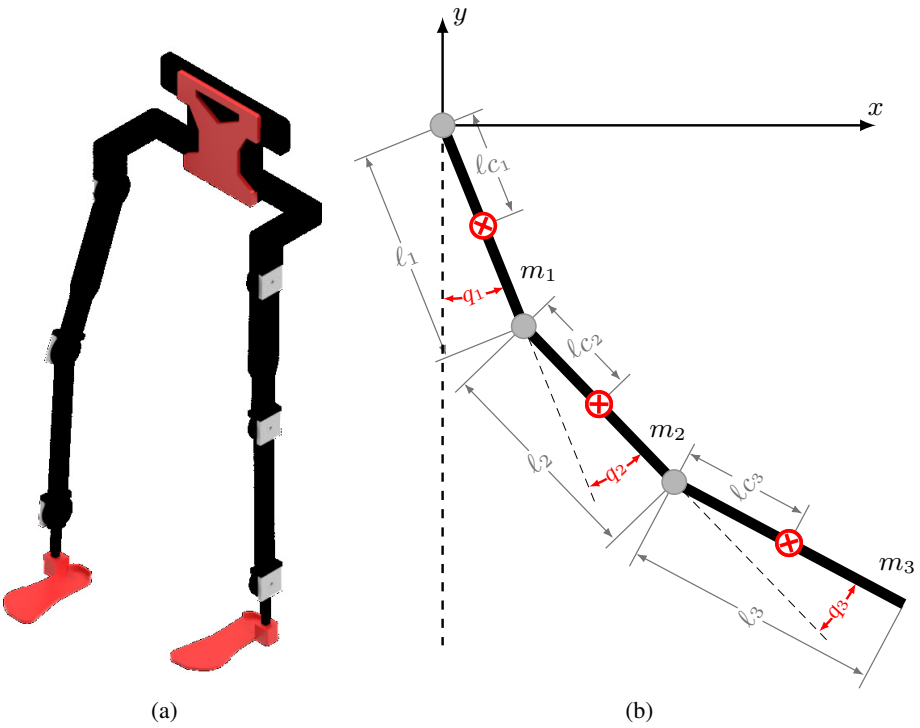


Fig. 1. (a) Mechanical design of a 6 DOF exoskeleton. (b) Rigid body diagram considered for Euler-Lagrange mathematical modelling.

In the specific case of disabilities in the lower limbs, they may originate from congenital anomalies [24], chronic conditions [4], or injuries [23, 12]. Overall statistics show that disabilities related to neuromuscular pathologies of lower limbs are prevalent. Among these medical conditions is the occurrence of an ischemic or hemorrhagic incident, commonly referred to as a stroke, which occurs when a thrombus obstructs or constricts an artery responsible for supplying blood to the brain.

Some lifestyle related risk factors are being overweight or obese, physically inactive or drinking alcoholic beverages in excess, meanwhile, medical risk factors encompass elevated blood pressure, high cholesterol, diabetes, among other conditions, with one of the prevailing complications being paralysis or impaired muscle movement [14]. To improve the quality of life of a person with such a complication, it is necessary to provide treatment options and therapies so that affected patients can successfully reintegrate into their daily activities.

One tool that may be useful in physical rehabilitation programmes is exoskeletons [19]. The development of mechatronic exoskeletons involves at least the following stages: mechanical design (number of degrees of freedom, kinematic and dynamic capabilities, mechanical constraints), instrumentation (sensor selection, actuators, processing system), control (control modules or algorithms defining the automatic behaviour of the device) [21].

Algorithm 1 PSO pseudocode.

Generate population
for $t = 1$: last generation **do**
  for $i = 1$: population size **do**
    if $f(X_{i,k}(t)) < f(p_i(t))$ **then** $f(p_i(t)) = f(X_{i,k}(t))$
      $f(G_{\text{best}}(t)) = \min(f(p_i(t)))$
    end if
    for $k = 1$ **to** problem dimension **do**
      $V_{i,k}(t+1) = wV_{i,k}(t) + c_1\ r_1(P_{\text{best}} - X_{i,k}(t)) + c_2\ r_2(G_{\text{best}} - X_{i,k}(t))$
      $X_{i,k}(t+1) = X_{i,k}(t) + V_{i,k}(t+1)$
      if $V_{i,k}(t+1) > v_{\text{max}}$ **then** $V_{i,k}(t+1) = v_{\text{max}}$
      else if $V_{i,k}(t+1) < v_{\text{min}}$ **then** $V_{i,k}(t+1) = v_{\text{min}}$
      end if
      if $X_{i,k}(t+1) > x_{\text{max}}$ **then** $X_{i,k}(t+1) = x_{\text{max}}$
      else if $X_{i,k}(t+1) < x_{\text{min}}$ **then** $X_{i,k}(t+1) = x_{\text{min}}$
      end if
    end for
  end for
end for

There is currently a wide variety of exoskeleton concept tests, including some commercial developments, such as the Indego, ReWalk, HAL, Exo-H3 and Ekso GT [19]. Commonly developments include four acting degrees of freedom (hip and knee) [10]. This imposes the challenge of extending the mechanical design to have exoskeletons that consider motion acting on the three main joints of the lower extremities, namely: hip, knee and ankle.

There are direct antecedents on the design and control of exoskeletons as reported in [22] and [2]; however, there are still many questions to be solved in terms of energy modulation of such devices. When discussing exoskeleton control strategies, it is essential to consider the three distinct levels involved: high level, medium level, and low level.

The high-level control pertains to land identification, where the focus lies on identifying and assessing the terrain. The medium-level control is concerned with the operation mode, determining the appropriate mode of operation based on the given circumstances. Lastly, the low-level control aims to achieve the desired torque or position for the articulations involved [1].

This research work develops the dynamic model from the Euler Lagrange equations based in an exoeskeleton model and proposes a PID controller [16] tuned using optimization tools such as swarm intelligence algorithms [20] applied to finding the appropriate gains to solve the low level control problem.

Comparison between two methods shows that objective function could be minimized to obtain best gains values, therefore, intelligent method calculates better performance values in terms of angular position error, so considering the calculated gains, the low-level control has been improved to accomplish with trajectory tracking problem.

Table 1. Dynamic model parameters.

Parameter	Value
Mass (m)	5
Length (ℓ)	0.45
Center of mass length (ℓ_c)	0.01
Inertia moment (I)	0.1595
Viscous friction coefficient (b)	0.17
Coulomb friction coefficient (f_c)	0.45
Gravity (g)	9.8

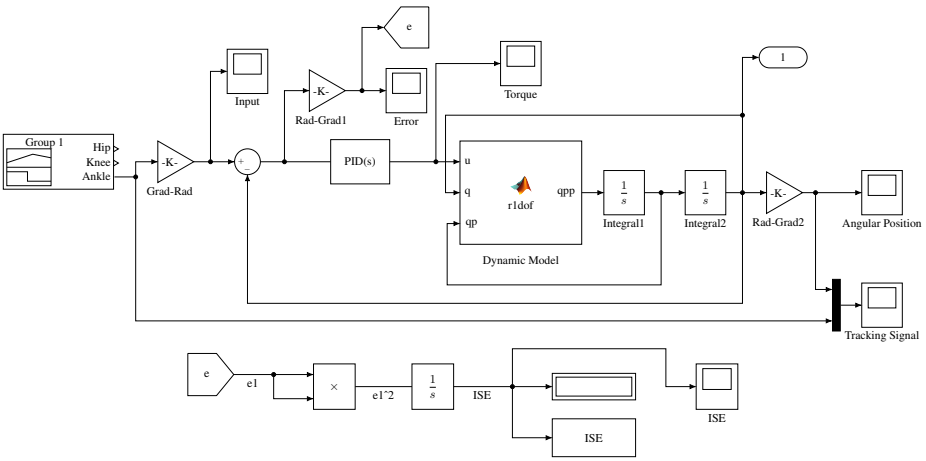


Fig. 2. Diagram developed in MATLAB-Simulink to obtain the trajectory tracking, trajectory error and torque from the three different input signals.

2 Exoeskeleton Mathematical Model

The mechanical design of the exoskeleton seeks to reproduce the joint movement of the lower extremities of the human body as shown in Figure 1a. The resulting mechanism is planar and its movement can be mathematically modeled with the Euler-Lagrange methodology for planar robots.

Because the mechanism reproduces the movement of the lower limb extremities, the model is proposed as a pair of open chains of three degrees of freedom each as shown in Figure 1b. This section describes the methodology for obtaining the dynamic model of one of the open chains, assuming it is the same methodology for two chains.

2.1 Euler-Lagrange Equations

The Euler Lagrange equations describes the dynamics of the articular positions and velocities of a mechanical system and is given by [9]:

$$\tau = \frac{d}{dt}\left[\frac{\partial L(q,\ \dot{q})}{\partial \dot{q}}\right] - \left[\frac{\partial L(q,\ \dot{q})}{\partial q}\right] + f_f(f_e,\ \dot{q}), \tag{1}$$

Table 2. PSO parameters.

Parameter	Value
Cognitive factor (c_1)	1.2
Social factor (c_2)	0.12
Inertia weight (w)	0.9
No. of particles (n)	30
Max iteration	30
Dimension (dim)	3
Random values ($r_1,\ r_2$)	rand(dim, n)
Initial conditions ($K_p,\ K_i,\ K_d$)	25, 5, 5

Table 3. Objective function values considering heuristic method.

Parameter	Value
Hip	11.13
Knee	15.33
Ankle	9.936

where, $\tau = [\tau_1 \quad \tau_2 \quad \tau_3]^T$ is the joint pairs vector, $q = [q_1 \quad q_2 \quad q_3]^T$ is the angular positions vector, $\dot{q} = [\dot{q_1} \quad \dot{q_2} \quad \dot{q_3}]^T$ is the angular velocities vector, $L(q,\ \dot{q})$ is the Lagrangian function and $f_f(f_e,\ \dot{q})$ is a friction function. The Lagrangian function is given by the following:

$$L(q,\ \dot{q}) = K_i(q,\ \dot{q}) - U_i(q),\ \ i = 1, 2, 3, \tag{2}$$

where, $K_i(q,\ \dot{q})$ is the kinetic energy and $U_i(q)$ is the potential energy. The kinetic energy of each bond of a mechanism is defined as:

$$K_i(q, \dot{q}) = \frac{1}{2}\left[m_i\ v_i^T\ v_i + I_i\ \dot{q}^2\right], \tag{3}$$

where, m_i is the mass of the link to be analyzed, v_i is the linear speed of the link and I_i is the moment of inertia tensor of the link. On the other hand, potential energy is defined as:

$$U_i(q) = m_i\ g\ h_i, \tag{4}$$

where, g is the ground gravity constant and h_i is the current height of the link with respect to the ground. To construct the equation of motion the equations (3) and (4) are replaced in the equation (2), the corresponding derivatives are calculated and thus the equation (1) is obtained.

Friction terms are not considered for this problem. A common way to represent the result of evaluating equation (1) is the so-called dynamic model in matrix form defined as [17]:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\ \dot{q} + G(q), \tag{5}$$

where, $\ddot{q} = [\ddot{q_1} \quad \ddot{q_2} \quad \ddot{q_3}]^T$ refers to the vector of angular accelerations, $M(q)$ is the inertial matrix, $C(q,\ \dot{q})$ is the matrix of centripetal forces and the gravity pair vector,

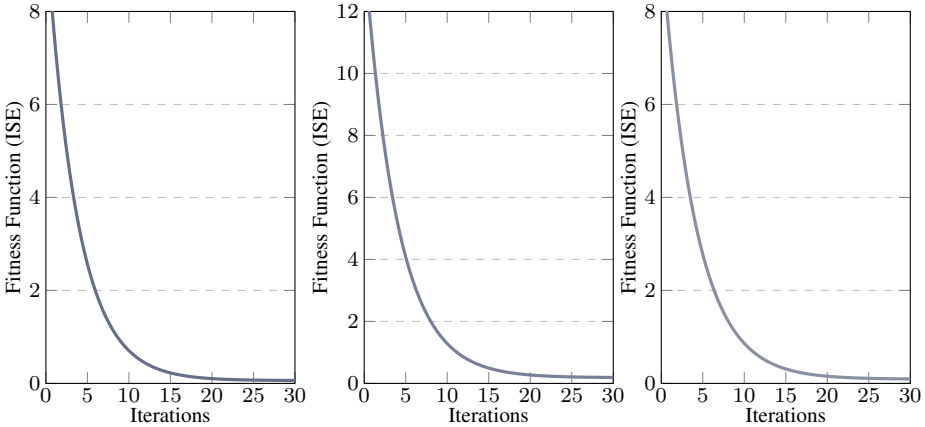


Fig. 3. (a) Fitness objective function for the hip trajectory, (b) fitness objective function for the knee trajectory (c) fitness objective function for the ankle trajectory.

denoted as $G(q)$:

$$M(q) = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix}, \tag{6}$$

where:

$$M_{11} = I_1 + I_2 + I_3 + \ell_1^2\, m_2 + \ell_1^2\, m_3 + \ell_2^2\, m_3 + \ell_{c_1}^2\, m_1 + \ell_{c_2}^2\, m_2 + \ell_{c_3}^2\, m_3 + 2\,\ell_1\,\ell_{c_3}\, m_3 \operatorname{Cos}(q_2 + q_3) + 2\,\ell_1\,\ell_2\, m_3 \operatorname{Cos}(q_2) + 2\,\ell_1\,\ell_{c_2}\, m_2 \operatorname{Cos}(q_2) + 2\,\ell_2\,\ell_{c_3}\, m_3 \operatorname{Cos}(q_3),$$

$$M_{12} = I_2 + I_3 + m_3\Big(\ell_2^2 + 2\,\ell_2\,\ell_{c_3} \operatorname{Cos}(q_3) + \ell_1\,\ell_2 \operatorname{Cos}(q_2) + \ell_{c_3}^2 + \ell_1\,\ell_{c_3} \operatorname{Cos}(q_2 + q_3)\Big) + \ell_{c_2}\, m_2\Big(\ell_{c_2} + \ell_1 \operatorname{Cos}(q_2)\Big),$$

$$M_{13} = I_3 + \ell_{c_3}\, m_3\Big(\ell_{c_3} + \ell_1 \operatorname{Cos}(q_2 + q_3) + \ell_2 \operatorname{Cos}(q_3)\Big),$$

$$M_{21} = I_2 + I_3 + m_3\Big(\ell_2^2 + 2\,\ell_2\,\ell_{c_3} \operatorname{Cos}(q_3) + \ell_1\,\ell_2 \operatorname{Cos}(q_2) + \ell_{c_3}^2 + \ell_1\,\ell_{c_3} \operatorname{Cos}(q_2 + q_3)\Big) + \ell_{c_2}\, m_2\Big(\ell_{c_2} + \ell_1 \operatorname{Cos}(q_2)\Big),$$

$$M_{22} = I_2 + I_3 + \ell_{c_2}^2\, m_2 + m_3\Big(\ell_2^2 + 2\,\ell_2\,\ell_{c_3} \operatorname{Cos}(q_3) + \ell_{c_3}^2\Big),$$

$$M_{23} = I_3 + m_3\Big(\ell_{c_3}^2 + \ell_2\,\ell_{c_3} \operatorname{Cos}(q_3)\Big),$$

$$M_{31} = I_3 + \ell_{c_3}\, m_3\Big(\ell_{c_3} + \ell_1 \operatorname{Cos}(q_2 + q_3) + \ell_2 \operatorname{Cos}(q_3)\Big),$$

$$M_{32} = I_3 + m_3\Big(\ell_{c_3}^2 + \ell_2\,\ell_{c_3} \operatorname{Cos}(q_3)\Big),$$

$$M_{33} = I_3 + \ell_{c_3}^2\, m_3.$$

$$C(q,\dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix}, \tag{7}$$

where:

$$\begin{aligned}
C_{11} =& \quad 0,\\
C_{12} =& -2\,\ell_1\,\dot{q_1}\Big(\ell_2\,m_3\,\mathrm{Sen}(q_2)+\ell_{c_2}\,m_2\,\mathrm{Sen}(q_2)+\ell_{c_3}\,m_3\,\mathrm{Sen}(q_2+q_3)\Big)-\ell_1\,\dot{q_2}\Big(\ell_2\,m_3\\
&\quad \mathrm{Sen}(q_2)+\ \ell_{c_2}\,m_2\,\mathrm{Sen}(q_2)+\ell_{c_3}\,m_3\,\mathrm{Sen}(q_2+q_3)\Big)-\ell_1\,\ell_{c_3}\,m_3\,\dot{q_3}\,\mathrm{Sen}(q_2+q_3),\\
C_{13} =& -\ \dot{q_1}\Big(2\,\ell_1\,\ell_{c_3}\,m_3\,\mathrm{Sen}(q_2+q_3)+2\,\ell_2\,\ell_{c_3}\,m_3\,\mathrm{Sen}(q_3)\Big)-m_3\,\dot{q_2}\Big(2\,\ell_2\,\ell_{c_3}\,\mathrm{Sen}(q_3)\\
&+\ \ell_1\,\ell_{c_3}\,\mathrm{Sen}(q_2+q_3)\Big)-\ell_{c_3}\,m_3\,\dot{q_3}\Big(\ell_1\,\mathrm{Sen}(q_2+q_3)+\ell_2\,\mathrm{Sen}(q_3)\Big),\\
C_{21} =& \quad 0,\\
C_{22} =& -\dot{q_1}\Big(m_3\Big(\ell_1\,\ell_2\,\mathrm{Sen}(q_2)+\ell_1\,\ell_{c_3}\,\mathrm{Sen}(q_2+q_3)\Big)+\ell_1\,\ell_{c_2}\,m_2\,\mathrm{Sen}(q_2)\Big),\\
C_{23} =& -\ell_{c_3}\,m_3\Big(2\,\ell_2\,\dot{q_1}\,\mathrm{Sen}(q_3)+2\,\ell_2\,\dot{q_2}\,\mathrm{Sen}(q_3)+\ell_2\,\dot{q_3}\ \mathrm{Sen}(q_3)+\ell_1\,\dot{q_1}\,\mathrm{Sen}(q_2+q_3)\Big),\\
C_{31} =& \quad 0,\\
C_{32} =& -\ell_1\,\ell_{c_3}\,m_3\,\dot{q_1}\,\mathrm{Sen}(q_2+q_3),\\
C_{33} =& -\ell_{c_3}\,m_3\,\dot{q_1}\Big(\ell_1\,\mathrm{Sen}(q_2+q_3)+\ell_2\,\mathrm{Sen}(q_3)\Big)-\ell_2\,\ell_{c_3}\,m_3\,\dot{q_2}\,\mathrm{Sen}(q_3).
\end{aligned}$$

$$G(q) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix}, \tag{8}$$

where:

$$\begin{aligned}
G_1 &= g\,m_3\Big(\ell_2\,\mathrm{Sen}(q_1+q_2)+\ell_1\,\mathrm{Sen}(q_1)+\ell_{c_3}\,\mathrm{Sen}(q_1+q_2+q_3)\Big)+\\
&\quad g\,m_2\Big(\ell_{c_2}\,\mathrm{Sen}(q_1+q_2)+\ell_1\,\mathrm{Sen}(q_1)\Big)+2\,g\,\ell_{c_1}\,m_1\,\mathrm{Sen}(q_1),\\
G_2 &= g\,m_3\Big(\ell_2\,\mathrm{Sen}(q_1+q_2)+\ell_{c_3}\,\mathrm{Sen}(q_1+q_2+q_3)\Big)+g\,\ell_{c_2}\,m_2\,\mathrm{Sen}(q_1+q_2),\\
G_3 &= g\,\ell_{c_3}\,m_3\,\mathrm{Sen}(q_1+q_2+q_3).
\end{aligned}$$

3 Exoskeleton Control System

Dynamic models can be examined through the approach of both linear and nonlinear systems. In the context of control system tuning, a heuristic approach is often employed for linear systems, while intelligent algorithms are typically harnessed when dealing with nonlinear systems. The heuristic tuning method in control is based on empirically adjusting the parameters of a controller rather than using a purely analytical or theoretical approach.

Table 4. Objective function values considering intelligent method.

Parameter	Value
Hip	0.1074
Knee	0.2328
Ankle	0.1341

This approach is used when the system is complex and optimal controller parameter values cannot be calculated directly. Intelligent tuning methods for control are based on the use of artificial intelligence (AI) or machine learning (ML) techniques to automatically adjust and optimize the parameters of a control system [25]. To illustrate the operation of the proposed design in the reproduction of lower limb movements, it is proposed to solve a trajectory tracking control problem using a PID control scheme that calculates the torque for each of the exoskeleton joints.

3.1 Control Scheme

It is considered $q_d = [q_{d_1} \;\; q_{d_2} \;\; q_{d_3}]^T$ as the reference trajectories to be reproduced on each of the legs of the exoskeleton, then considering the angular position $q = [q_1 \;\; q_2 \;\; q_3]^T$, the position error is defined as the vector:

$$e(t) = \begin{bmatrix} q_{d_1}(t) - q_1(t) \\ q_{d_2}(t) - q_2(t) \\ q_{d_3}(t) - q_3(t) \end{bmatrix}, \tag{9}$$

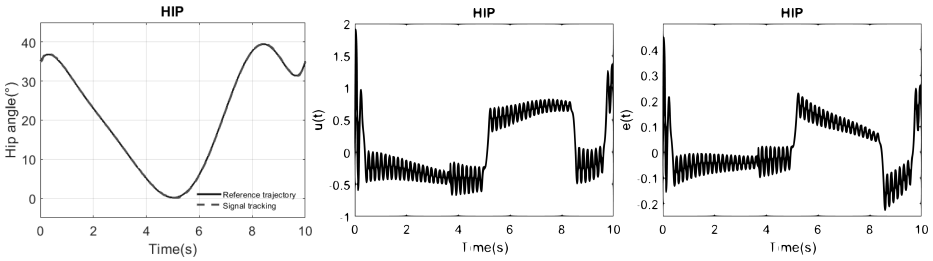
From the error vector $e(t)$, the PID law control [16] is defined as:

$$\tau = K_p\, e(t) + K_v\, \dot{e}(t) + K_i \int_0^t e(t)\, dt, \tag{10}$$

where $\tau = [\tau_1 \;\; \tau_2 \;\; \tau_3]^T$ are the pairs in the three joints of each leg and K_p, K_v y $K_i \in R^{3\times 3}$ are the defined positive gain matrices, called proportional, derivative and integral gain, respectively. To solve the control problem, the reference trajectories q_d are then required. In this work, the data reported in [18] about the angular positions of the hip, knee, and ankle (both legs) in study subjects who performed walking activities were used as reference trajectories.

3.2 Optimization of the Controller´s Gains

The numeric value of the gain matrices is calculated based on Particle Swarm Optimization algorithm (PSO), such that the closed-loop system response converges to the proposed input reference. PSO is an optimization technique inspired by the cooperative patterns observed in nature, such as the flocking of birds and the schooling of fish.



(a) (b) (c)

Fig. 4. (a) Trajectory tracking signal, (b) input torque and (c) trajectory error signal, of the hip joint.

Table 5. Calculated gains from PSO algorithm.

Joint	K_p	K_i	K_d
Hip	240.9220	122.7799	23.8527
Knee	210.6642	131.6928	3.45810
Ankle	241.7013	27.69170	17.6263

It is a metaheuristic approach that exhibits several advantages: PSO necessitates minimal or no assumptions regarding the optimization problem at hand, it does not depend on problem differentiability, and it can explore vast solution spaces. These attributes make PSO a potent tool for tackling multidimensional and intricate optimization problems.

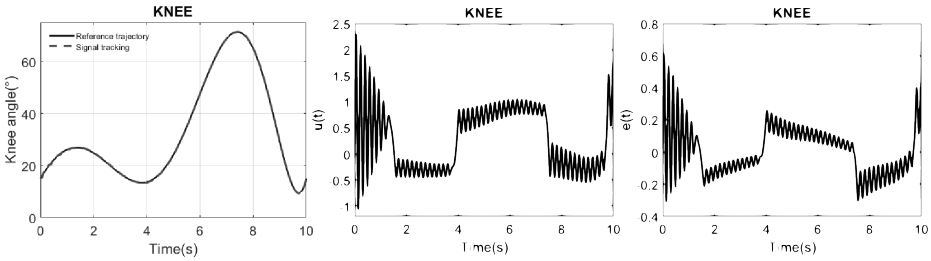
Within the framework of the PSO algorithm, the particles, representing potential solutions, navigate through the search space by aligning their movements with the current optimal particle. Essentially, at the k-th iteration, each particle p_i in the swarm possesses two attributes: its position denoted as $X(t)$ and its velocity denoted as $V(t)$.

The particle's motion is determined in terms of velocity that combines the influence of its own best position P_{best} and the collective best position of the complete swarm G_{best} inside the exploration domain:

$$V(t+1) = w\, V(t) + c_1\, r_1\ (P_{\text{best}} - X(t)) + c_2\, r_2\, (G_{\text{best}} - X(t))\,, \tag{11}$$

$$X(t+1) = X(t) + V(t+1). \tag{12}$$

More precisely, the speed of each individual particle undergoes an update using Equation (11). In this equation, the cognitive and social learning coefficients are represented by c_1 and c_2, respectively. The prescribed inertia weight is represented by w, while r_1 and r_2 denote random numbers generated within the range of 0 and 1 during each iteration. The particle's position can be updated using Equation (12). The iteration process concludes either when the predetermined number of iterations is attained or when the objective function $f(t)$ reaches a critical value [11].



(a) (b) (c)

Fig. 5. (a) Trajectory tracking signal, (b) input torque and (c) trajectory error signal, of the knee joint.

3.3 Objective Functions

Selecting the objective functions to assess the fitness of each particle stands as a pivotal step in the implementation of PSO [15]. Among the objective functions commonly employed are the Mean Squared Error (MSE), the Integral of Time multiplied by Absolute Error (ITAE), the Integral of Absolute Error (IAE), and the Integral of Squared Error (ISE) [7]. In this study, the objective function employed is ISE due to its minimal energy consumption in the context of energy control [13] and is described by the following equation:

$$\text{ISE} = \int_0^t e^2(t)\, dt, \tag{13}$$

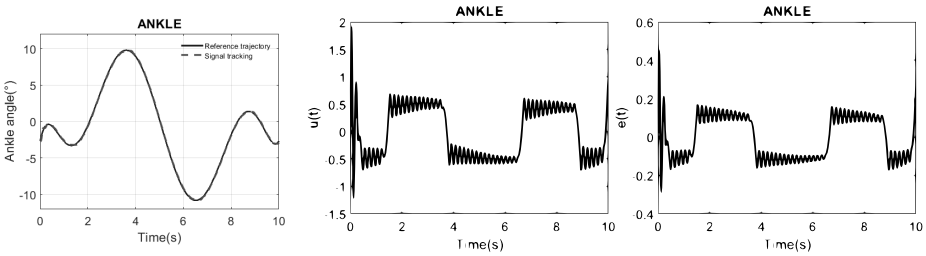
where $e(t)$ is the error signal and is based in Equation (9).

4 Experiment Design

The simulation of the exoskeleton experiment as shown in Figure 2, is developed in Matlab-Simulink considering a mathematical model of 1 DOF, the trajectory input signals, the PID control system and the PSO intelligent tuning method using ISE as objective function. The decision variables considered in the PSO algorithm are based on the minimization of the objective function ISE, when PSO finds a better population of particles, the value of the objective function is updated until the end of the simulation by restricting the iterations.

It is well known that the PID control law does not generate an equilibrium point of the closed-loop system with global asymptotic stability characteristics, it only has local asymptotic stability as long as the gains are positive defined, therefore, the algorithm has the constraint of only looking for positive gain values such that the closed-loop system response converges to the reference trajectories.

It is important to mention that ISE function global minimum value is 0. Thus, PSO algorithm works to optimize the objective function to zero that is equals to calculate a minimum trajectory tracking error.



(a) (b) (c)

Fig. 6. (a) Trajectory tracking signal, (b) input torque and (c) trajectory error signal, of the ankle joint.

Then, the optimization problem can be defined as follows:

$$e(t) = q_{di}(t) - q_i(t), \text{ such that, } \text{ISE} = \int_0^t e^2(t)\, dt = 0 \quad (\text{Global minimum}), \quad (14)$$

considering that, K_p, K_i and $K_d \in R^+$. Thus:

$$\tau = K_p\, e(t) + K_v\, \dot{e}(t) + K_i \int_0^t e(t)\, dt, \text{ converge to trajectory signals.} \quad (15)$$

As mentioned in subsection 3.1, the trajectory input signals was taken from [18] and the acceleration of the 1 DOF dynamical model is calculated as follow:

$$\ddot{q} = \frac{\left[\tau - m\, g\, \ell_c \operatorname{Sen} q - b\, \dot{q} - f_c\ \operatorname{signo}(\dot{q})\right]}{\left[\ell_c^2 + I\right]}. \quad (16)$$

The parameters used to simulate the dynamic model are shown in Table 1. The experiment parameters to simulate the PSO are presented in Table 2 and as mentioned in subsection 3.3, ISE is used as objective function based on the error signal from the closed-loop system. Table 3 shows the objective function calculated values based on the heuristic method. Also, gain values from heuristic method were used as initial conditions for the intelligent method. Objective function for the three joints during the iterative process of the intelligent method is presented in Figure 3 and the calculated values are presented in Table 4.

5 Results

Having considered the previous work, this section presents the results of the dynamic model from the Euler Lagrange motion equations and the solution of the trajectory tracking control problem using a PID schema based in PSO to find the best gain values based on ISE objective function. The calculated gain values are presented in Table 5. The results of tracking, input torque and error for joints are shown in Figures 4, 5 and 6.

6 Conclusions

Utilizing conventional methods to implement a controller for a multiple joint structure proves to be challenging and lacks precision. Integrating intelligent swarm optimization techniques, such as PSO, presents a viable approach for fine-tuning PID controllers. As mentioned in subsection 3.3, there is a range of objective functions available for consideration.

However, within the context of energy control, ISE was selected as the objective function to be optimized by the PSO algorithm for tuning the PID controller gains. Results shows that the tuned controller scheme generate adequate torques to accomplish the trajectory tracking problem. The intelligent method calculates better performance values in terms of angular position error, so considering the generated gains, the low-level control was improved compared with the heuristic method.

References

1. Baud, R., Manzoori, A. R., Ijspeert, A., Bouri, M.: Review of control strategies for lower-limb exoskeletons to assist gait. Journal of NeuroEngineering and Rehabilitation, vol. 18, no. 1 (2021) doi: 10.1186/s12984-021-00906-3
2. Biao, L., Youwei, L., Xiaoming, X., Haoyi, W., Longhan, X.: Design and control of a flexible exoskeleton to generate a natural full gait for lower-limb rehabilitation. Journal of Mechanisms and Robotics, vol. 15, no. 1 (2022) doi: 10.1115/1.4054248
3. Brault, M.: Americans with disabilities: Current population reports (2010) www.census.gov /library/publications/2012/demo/p70-131.html
4. CDC: Division of birth defects and developmental disabilities, data and statistics (2020) www.cdc.gov
5. Chan, M., Zoellick, R. B.: World report on disability (2011) iris.who.int/bitstream/handle/ 10665/44575/9789240685215_eng.pdf?sequence=1
6. Eurostat: Population by type of basic activity difficulty, sex and age (2019) ec.europa.eu/eur ostat/databrowser/view/HLTH_DP040/default/table?lang=en&category=hlth.hlth_dsb.hlth_d sb_prv
7. Griffin, I., Bruton, J.: On-line PID controller tuning using genetic algorithms Dublin City University (2003)
8. Gutiérrez, J. P., Rivera-Dommarco, J., Shamah-Levy, T., Villalpando-Hernández, S., Franco, A., Cuevas-Nasu, L., Romero-Martínez, M., Hernández-Ávila, M.: Encuesta nacional de salud y nutrición. Resultados nacionales (2012) ensanut.insp.mx/encuestas/ensanut201 2/doctos/informes/ENSANUT2012ResultadosNacionales.pdf
9. Kelly, R.: Control de movimiento de robots manipuladores. Pearson Education (2003)
10. Kim, H., June-Shin, Y., Kim, J.: Design and locomotion control of a hydraulic lower extremity exoskeleton for mobility augmentation. Mechatronics, vol. 46, pp. 32–45 (2017) doi: 10.1016/j.mechatronics.2017.06.009
11. Liu, J., Fang, H., Xu, J.: Online adaptive PID control for a multi-joint lower extremity exoskeleton system using improved particle swarm optimization. Machines, vol. 10, no. 1, pp. 21 (2021) doi: 10.3390/machines10010021
12. Mackay, J., Mensah, G.: World health organization: The atlas of heart disease and stroke (2004) apps.who.int/iris/handle/10665/43007

13. Maghfiroh, H., Saputro, J. S., Hermanu, C., Ibrahim, M. H., Sujono, A.: Performance evaluation of different objective function in PID tuned by PSO in DC-motor speed control. In: Proceedings of the IOP Conference Series: Materials Science and Engineering, vol. 1096 (2021) doi: 10.1088/1757-899x/1096/1/012061
14. Mayo Clinic: Accidente cerebrovascular (2022) www.mayoclinic.org
15. Mirzal, A., Yoshii, S., Furukawa, M.: PID parameters optimization by using genetic algorithm. Journal of International Science and Technology Conference, vol. 8, pp. 34–43 (2006) doi: 10.48550/arXiv.1204.0885
16. Ogata, K.: Ingeniería de control moderna. Pearson Education (2010)
17. Reyes, F.: Robótica: control de robots manipuladores. Alfa Omega (2011)
18. Ribeiro, T. S., de-Sousa, A. C., de-Lucena, L. C., Santiago, L. M. M., Lindquist, A. R. R.: Does dual task walking affect gait symmetry in individuals with Parkinson's disease? European Journal of Physiotherapy, vol. 21, no. 1, pp. 8–14 (2018) doi: 10.1080/21679169.2018.1444086
19. Sharifi, M., Mehr, J. K., Mushahwar, V. K., Tavakoli, M.: Adaptive CPG-based gait planning with learning-based torque estimation and control for exoskeletons. IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 8261–8268 (2021) doi: 10.1109/lra.2021.3105996
20. Soleimani-Amiri, M., Ramli, R., Ibrahim, M. F., Abd-Wahab, D., Aliman, N.: Adaptive particle swarm optimization of pid gain tuning for lower-limb human exoskeleton in virtual environment. Mathematics, vol. 8, no. 11 (2020) doi: 10.3390/math8112040
21. Tibaduiza-Burgos, D. A., Aya-Parra, P. A., Anaya-Vejar, M.: Exoesqueleto para rehabilitación de miembro inferior con dos grados de libertad orientado a pacientes con accidentes cerebrovasculares. INGE CUC, vol. 15, no. 2, pp. 36–47 (2019) doi: 10.17981/ingecuc.15.2.2019.04
22. Tovar-Estrada, M., Rodriguez-Liñan, A., Quiroz, G.: Implementation of a scale-lab lower-limb exoskeleton with motion in three anatomical planes. Cybernetics and Systems, vol. 50, no. 6, pp. 516–538 (2019) doi: 10.1080/01969722.2019.1630565
23. World Health Organization: Spinal cord injury (2013) www.who.int/news-room/fact-sheets/detail/spinal-cord-injury
24. World Health Organization: Congenital anomalies (2022) www.who.int/health-topics/congenital-anomalies#tab=tab_1
25. Yu, W.: PID Control with Intelligent Compensation for Exoskeleton Robots. Elsevier (2018) doi: 10.1016/C2016-0-04547-3

ISSN 1870-4069

Assistive Technologies for American Sign Language Users: A Systematic Mapping Study

Miguel Avila-Cabrera, Antonio Aguileta-Güemez,
Jorge Rios-Martinez, Jorge Reyes-Magaña

Universidad Autónoma de Yucatán,
Yucatán,
Mexico

miguel@avila.id, {aaguilet, jorge.rios,
jorge.reyes}@correo.uady.mx

Abstract. This systematic mapping study provides a comprehensive overview of research on assistive technologies for American Sign Language (ASL). It utilized the Scopus database, employing a carefully formulated search string and inclusion/exclusion criteria. The findings indicate that research in this field is primarily conducted in the northern region, particularly the United States. While the primary focus lies on interpreters and translators, there is a noticeable scarcity of investigations into harnessing mainstream technologies for the benefit of ASL users through AI-powered solutions like Personal Assistants. The study emphasizes the need for further advancements in enhancing accessibility for different contexts and impairments. It contributes to understanding the research landscape and identifies avenues for future research and development in ASL assistive technologies.

Keywords: AI assistance, assistive technologies, speech impairment, human-computer interaction, interaction design, accessibility.

1 Introduction

Assistive technologies for individuals with disabilities cover a broad spectrum of innovative solutions that strive to augment independence, accessibility, and quality of life [61]. Moreover, assistive technologies contribute to fostering inclusivity, engagement, and equitable opportunities in educational, occupational, and social contexts [11]. American Sign Language (ASL) is a visual-gestural language that is primarily utilized by the Deaf community in the United States and certain regions of Canada [41]. ASL is renowned for its linguistic complexity and serves as a vital medium enabling Deaf individuals to partake in conversations, share narratives and express emotions.

In the realm of assistive technologies for individuals with speech impairments, artificial intelligence (AI) has assumed a central role [9]. AI-driven solutions leverage natural language processing capabilities to address the communication challenges faced by those with speech disabilities. These encompass a spectrum of applications, notably including text-to-speech (TTS) synthesis, automatic speech recognition (ASR), and augmentative and alternative communication (AAC) systems [9].

AAC systems harness AI to anticipate and suggest words or phrases based on user input, streamlining the communication process. Rooted in AI and machine learning algorithms, these technologies exhibit a trajectory of ongoing enhancement in terms of accuracy and user-friendliness [48]. Their potential to empower individuals with speech impairments in achieving more effective communication is substantial.

Researchers and developers within this domain are actively channeling AI innovations to elevate the accessibility and efficacy of assistive technologies tailored for speech-impaired individuals, thus culminating in tangible enhancements in their overall quality of life [61]. Assistive technologies for American Sign Language (ASL) and individuals with deafness or speech impairments encompass innovative solutions that promote communication and inclusivity [48].

These technologies aim to bridge the communication gap between ASL users and those unfamiliar with the language [48]. For instance, video relay services, where Artificial Intelligence driven real-time interpretation facilitates seamless communication between ASL users and speakers of spoken languages through video calls [55, 4] and Personal Assistants triggered by gesture recognition powered by AI [16, 42, 15]. Sign language recognition systems employ computer vision and machine learning algorithms to interpret ASL gestures and translate them into text or speech.

To gain a deeper understanding of assistive technologies related to American Sign Language (ASL), we conducted a systematic mapping study. This study enabled us to examine the research on assistive technologies for ASL within a specific context. Systematic mapping studies help create a structured framework for exploring a research topic and present a concise visual summary of the identified findings [31, 52].

To establish a comparative analysis with other relevant studies, including systematic mapping studies in various assistive technology fields [46, 35, 23, 1]. In congruence with previous scholarly endeavors, the present study exhibits noteworthy resemblances concerning the employed assistive technology and the methodological approach employed for their application, our investigation focuses specifically on exploring in-depth inquiries related to user experience and user context.

Our main objective is to develop and offer assistive technologies that are specifically tailored to individual user requirements. We organized this Systematic Mapping Study is structured as follows. The first section aims to describe the methodology. In Results section, we showcase the results of our study. In Discussions section we discuss over the findings and finally we summarize the research in the Conclusions section.

2 Research Method

Our research employed the systematic mapping study approach, a recognized and rigorous scientific methodology [31, 52]. This research method shares similarities with the well-established Systematic Literature Review (SLR) approach commonly used in scientific research [30]. The process of conducting a mapping study follows a systematic and rigorous framework, involving three fundamental activities:

– **Planning.** During this stage, the mapping study protocol is meticulously developed, involving a rigorous and iterative process to establish the overarching plan of the mapping study.

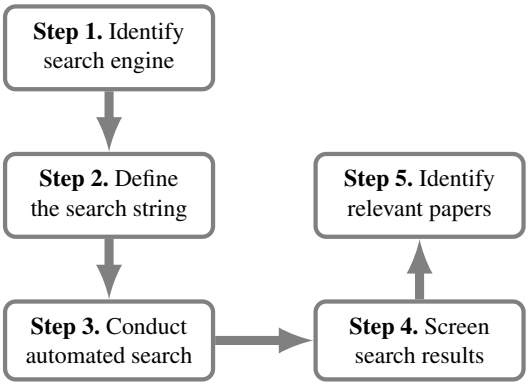


Fig. 1. Sequence of steps in the proposed mapping search procedure.

– **Execution.** During this phase, the mapping study protocol is executed, entailing the implementation of the predefined search string on the designated sources. The retrieved documents are then systematically evaluated based on the predetermined inclusion and exclusion criteria.
– **Reporting.** This phase involves reporting the mapping study findings and ensures the transparency, credibility, and reproducibility of the study's findings, contributing to the advancement of knowledge in the respective research area.

2.1 Research Question Definition

Our study aims to provide a comprehensive overview of the current state of the art in Assistive Technologies for American Sign Language (ASL). To achieve this, we have formulated research questions that align with our overall objective. These questions guide the identification and categorization of the existing research in accordance with our defined goal:

– **RQ1:** What is the distribution of research papers about Assistive Technologies for American Sign Language Users categorized by country?
– **RQ2:** What is the distribution of research papers about Assistive Technologies for American Sign Language Users categorized by year?
– **RQ3:** What is the distribution of the research papers about Assistive Technologies for American Sign Language Users categorized by the Impairment Type?
– **RQ4:** What is the distribution of the research papers about Assistive Technologies for American Sign Language Users categorized by the Assistive Technology used?

2.2 Identification and Selection of Sources

We utilized the widely recognized Scopus database as our primary information source for this study. Scopus offers an extensive collection of scientific literature, granting access to a diverse range of publications.

Table 1. Search string defined for the systematic mapping study.

Search String
(“AI Assistance” OR “Personal Assistant” OR “Virtual Assistant” OR “Assistive Technologies”) AND TITLE-ABS-KEY (“ASL” OR “American Sign Language”) AND (“Human-computer Interaction” OR “Accessibility” OR “HCI”)

Figure 1 illustrates the overall system of the Systematic Mapping Study methodology we followed. We developed the search string for our study by extracting relevant terms from the research questions. These terms were combined using logical operators like “AND” and “OR” to refine the search and identify relevant literature.

Table 1 presents the resulting search string, highlighting that the majority of the terms focus on assistive technologies. After choosing the search source and defining the search string, we established specific inclusion (IC) and exclusion (EC) criteria for the selection of primary studies. Table 2 provides a concise overview of the inclusion and exclusion criteria utilized in our study.

2.3 Execution

After finalizing the inclusion and exclusion criteria, we executed the search string on the Scopus database, specifically, the database query was executed on May 14, 2023. To conduct this evaluation, we thoroughly examined the titles, abstracts, and keywords of all the retrieved documents.

In some cases, a detailed screening of the entire paper was necessary to assess its eligibility for inclusion. Table 3 presents the number of relevant papers. To organize the information from the selected relevant papers systematically, we created a structured template. This template included specific fields to capture essential details, ensuring consistency and facilitating analysis.

3 Results

In this section, the finding results and analyses of the paper categorization are presented. According to the findings, we address each of the four research questions (RQs) defined in the Research Method section.

3.1 What is the Distribution of the Research Papers About Assistive Technologies for American Sign Language Users Categorized by Country? (RQ1)

The first question aims to identify the number of published relevant papers across the world. We categorized the papers according to the author’s affiliation and organized them based on their respective country. If the papers were written by two or more authors of different countries these papers were duplicated and accounted for in each country to which the authors belong. Figure 2 shows a choropleth map of the distribution of papers across the world.

Table 2. Exclusion and inclusion criteria.

Criteria	Description
IC1	Include English papers.
IC2	Include papers that contain the search String terms.
IC3	Include papers that maintain relationship with the keywords.
EC1	Exclude papers that do not contain the characteristics mentioned above.

We identified one country with the greater number of published papers (32) [44, 34, 55, 16, 18, 45, 24, 42, 33, 15, 5, 54, 8, 14, 19, 10, 51, 29, 20, 60, 28, 40, 27, 25, 26, 53, 56, 21, 39, 59, 38, 22]: United States of America. The remaining published papers (18) belong to 17 countries: Canada (1) [32], Cyprus (1) [62], Germany (1) [13], India (2) [17, 57], Norway (1) [57], Italy (2) [12, 2], Korea (2) [37, 36], Philippines (2) [3], Spain (1) [43], Thailand (1) [50], Tunisia(3) [7, 49], France (1) [8], Netherlands (1) [8], UK (1) [6], Australia (1) [6]. This finding suggest that Assistive technologies for ASL research is of interest in a great variety of countries, tough most of the publications belong to America, more in detail to United States of America.

3.2 What is the Distribution of the Research Papers About Assistive Technologies for American Sign Language Users Categorized by Year? (RQ2)

This research question aims to identify the number of published papers by year. We categorized the papers according to the year of publication. Figure 3 shows a Scatter plot of the distribution of papers across the years. We identified two years with the greater number of published papers (18): In the period from 2021 to 2022 [37, 44, 62, 34, 55, 16, 18, 13, 32, 45, 24, 50, 42, 33, 15, 3, 5, 54].

The remaining published papers (29) belong to 14 periods of year: 2023 (1) [17], 2019 (4) [8, 12, 7, 2], 2018 (1) [36], 2017 (2) [14, 19], 2016 (3) [10, 51, 29], 2015 (4) [20, 43, 60, 28], 2014 (3) [40, 27, 25], 2013 (3) [26, 57, 49], 2012 (4) [53, 56, 6, 21], 2011 (1) [39], 2010 (1) [59], 2009 (1) [38], 2005 (1) [22]. This finding suggest that AI assistance for ASl research has been from interest in recent years from the period of 2021 to 2023 as most of the publications belong to that period.

3.3 What is the Distribution of the Research Papers About Assistive Technologies for American Sign Language Users Categorized by the Impairment Type? (RQ3)

This research question aims to identify the number of published papers by the Impairment they had focus on. We categorized the papers according to the Indiana University classification of Types of Impairment [58]. If the papers were written considering two or more impairment types we created a new category including the combination of these impairments.

Figure 4 shows a pie chart of the distribution of papers by impairment type. As observed in Figure 4 the two most common types of impairments are Speech and Hearing impairment independently (39) [17, 44, 62, 55, 16, 18, 13, 32, 45, 24, 50,

Table 3. Database search results.

Search Date	Document Results	Relevant Papers
05/14/2023	91	47

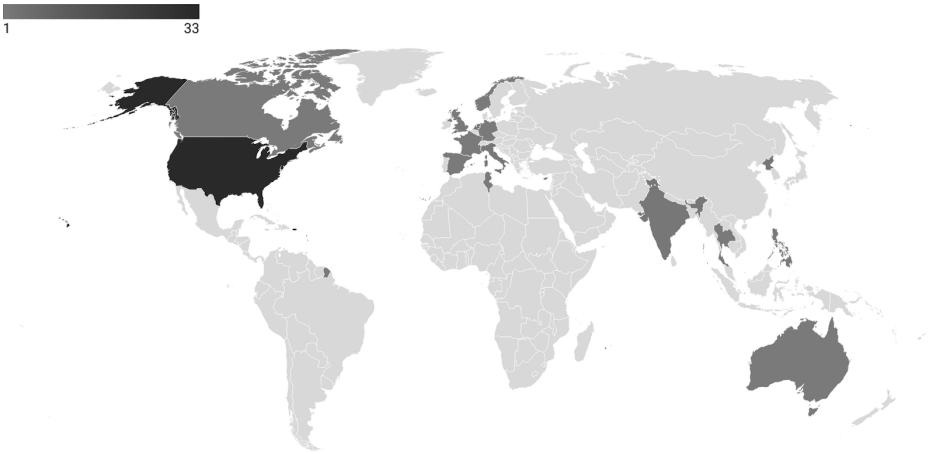


Fig. 2. Choropleth Map that highlights the papers published across the world.

42, 33, 3, 5, 54, 8, 12, 7, 2, 36, 14, 19, 10, 51, 29, 20, 43, 60, 28, 40, 27, 25, 26, 57, 49, 56, 38, 22]. We also observe that only one combination (Speech and Hearing) (8) [37, 34, 15, 53, 6, 21, 39, 59] is generated, as they both make use of the American Sign Language.

3.4 What is the Distribution of the Research Papers About Assistive Technologies for American Sign Language Users Categorized by the Assistive Technology Used (RQ4)

This research question aims to identify the number of published papers by the Assistive Technology they used or suggested to use.We categorized the papers according to the National Institute on Deafness and Other Communication Disorders [47]. Figure 5 shows a distribution chart of papers by assistive technology. Within the domain of personal assistants, our research explores various aspects.

We have identified noteworthy papers in this context, including those investigating command triggers, which aim to understand the factors that initiate and activate commands within personal assistant systems (3 papers) [16, 42, 15] it is worth noting these papers purposes a gesture recognition systems to provide the inputs.

Additionally, there are papers focusing on interpreters and translators, which are widely used assistive technologies to facilitate effective communication across language barriers (24 papers) [62, 18, 13, 45, 24, 5, 54, 12, 36, 19, 10, 29, 20, 60, 28, 57, 49, 53, 56, 6, 21, 39, 59, 22] within these papers there are some using gesture recognition and natural language processing empowered by AI. The scientific literature also encompasses innovative approaches proposed in captioning papers, shedding light

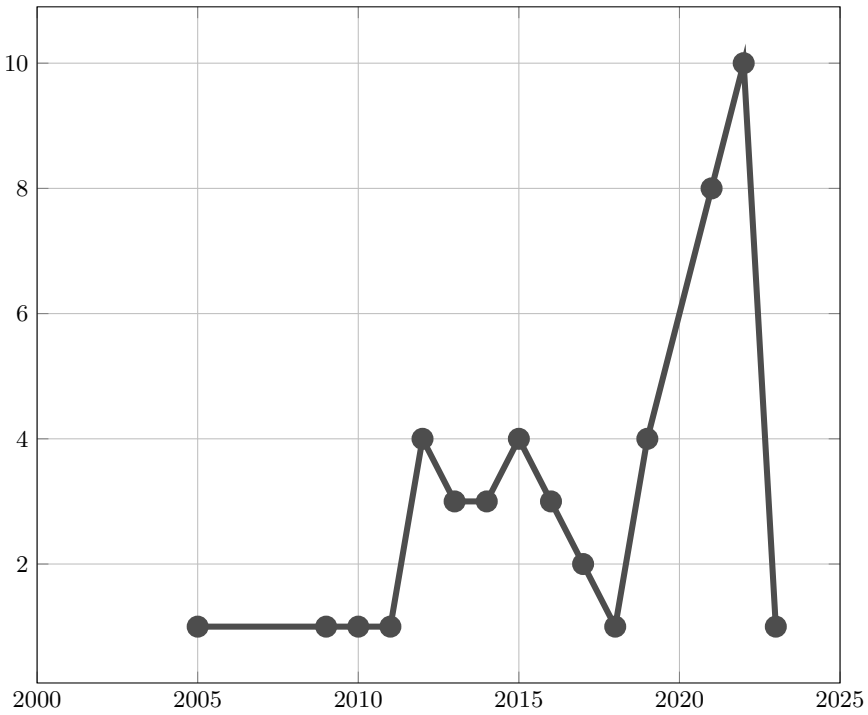


Fig. 3. Distribution of relevant papers published per year.

on ASL captioning, integration within video conferencing platforms, and the fusion of videos with automatic captioning (2 papers) [44, 55]. Furthermore, theoretical applications papers propose potential applications for assistive technology in this field (18 papers) [17, 37, 34, 32, 50, 33, 3, 8, 7, 2, 14, 51, 43, 40, 27, 25, 26, 38].

4 Discussions

The selected papers in this mapping study were authored by individuals from various countries, with a majority of them being published by authors affiliated with the United States of America. This indicates that the majority of research in this field comes from countries in the northern region, such as Canada, Germany, France, Norway, Italy, Korea, Spain, and the Netherlands. On the other hand, there is a noticeable lack of research from countries in the southern region, with only Jamaica contributing papers on the topic (RQ1).

In recent years, there has been an increased focus on Assistive Technologies for Speech Impairment and American Sign Language (ASL) users, particularly in the period of 2021 and 2022. This surge in attention coincides with the global outbreak of the COVID-19 pandemic, which some papers acknowledge as a contextual factor influencing research in this area (RQ2).

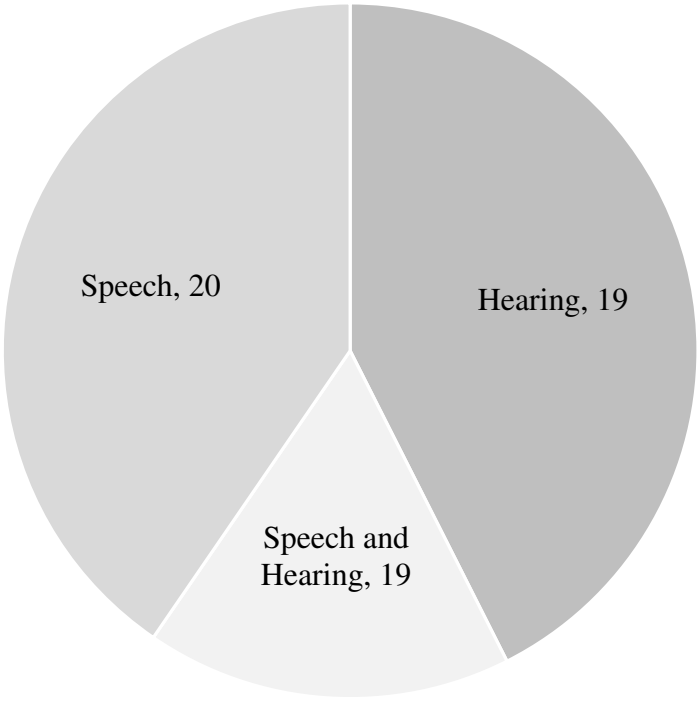


Fig. 4. Distribution of papers published by type of impairment.

Regarding the categorization of published papers based on impairment type, it is evident that hearing impairment, speech impairment, and a combination of both have received the most attention. This specialization in research indicates a recognition of the unique challenges and requirements associated with different types of impairments, with the aim of developing tailored solutions that cater to the specific needs of individuals with hearing or speech impairments (RQ3).

Among the published papers, a significant proportion (44.64%) focuses on research related to Interpreters and Translators, followed by Theoretical Applications, accounting for 37.5% of the papers. This distribution highlights the challenge of bridging the gap between theoretical concepts and practical implementation in fully harnessing the potential of assistive technologies for individuals with disabilities.

Notably, there are also research efforts focused on personal assistants and captioning tools. However, it is important to note that while many papers propose new technologies, their primary focus lies in enhancing communication for ASL users through the development and improvement of interpreters and translators.

The findings of this study unequivocally demonstrate that artificial intelligence (AI) plays a pivotal role in the development and advancement of assistive technologies. Specifically, AI technologies have become integral in facilitating gesture recognition for translation and interpretation purposes, as well as enhancing personal assistants' capabilities to provide relevant and valuable responses.

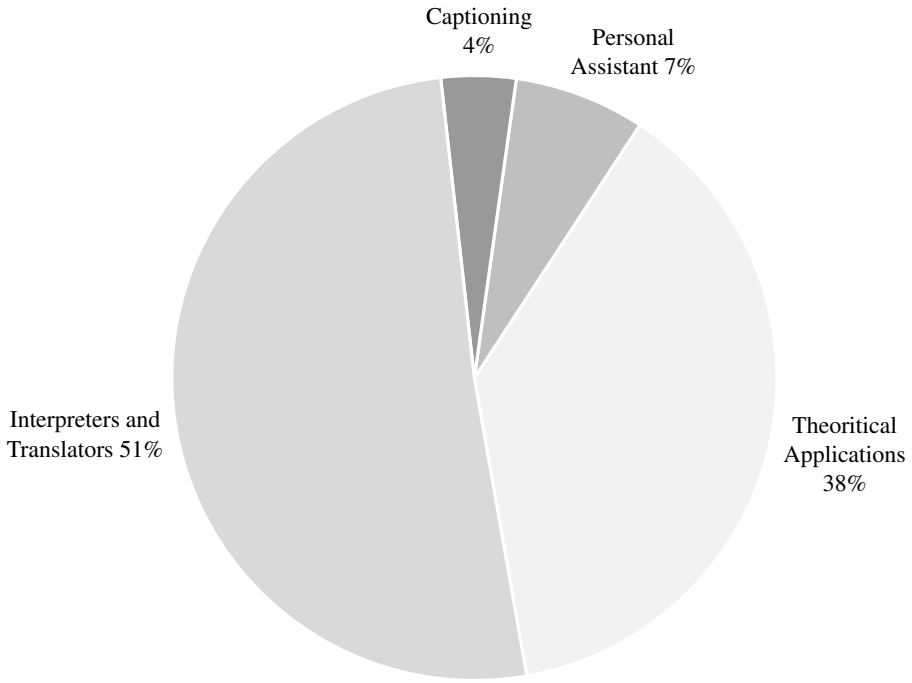


Fig. 5. Distribution of papers published by assistive technology.

The adaptation of technologies like personal assistants and captioning tools for this specific user group receives relatively less attention (RQ4). Despite the increased visibility of Assistive Technologies research for ASL users on an international scale, there is still a need for further advancements in this field. Our findings indicate that a significant proportion of published papers primarily focus on theoretical applications, with a noticeable lack of user experience (UX) research methods to validate their accessibility.

5 Conclusions

This study aims to comprehensively characterize the research landscape of assistive technologies for American Sign Language (ASL) through a systematic mapping study of relevant scientific and technical papers. The research area of assistive technologies for ASL users has gained significant relevance worldwide using AI as the main engine for these technologies. While many papers propose innovative technologies, their primary focus is on improving communication for ASL users, particularly through the development and refinement of interpreters and translators.

However, there is relatively less emphasis on adapting mainstream technologies for this specific user group such as personal assistants. Based on these findings, it can be concluded that there is room for improvement in enhancing the accessibility of these technologies for users across different contexts and impairments such as the integration

of mainstream technologies that could potentially be harnessed to cater to the needs of American Sign Language (ASL) users through the utilization of Artificial Intelligence (AI) methodologies. In forthcoming research endeavors, we intend to conduct a more comprehensive investigation into the state-of-the-art of assistive technologies. This endeavor will encompass an in-depth exploration of recent advances in the field, delving into novel research questions and facilitating a more profound analysis of pertinent research papers.

Such endeavors can encompass, among other possibilities, the exploration of novel research inquiries concerning the methodologies underpinning user-centered and accessibility aspects within the realm of assistive technologies. Furthermore, a meticulous delineation of each scholarly work is imperative to provide a comparative perspective and enhance the elucidation of the field.

References

1. Asghar, I., Cang, S., Yu, H.: A systematic mapping study on assitive technologies for people with dementia. In: 9th International Conference on Software, Knowledge, Information Management and Applications, pp. 1–8 (2015) doi: 10.1109/SKIMA.2015.7399989
2. Avola, D., Bernardi, M., Cinque, L., Foresti, G. L., Massaroni, C.: Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. IEEE Transactions on Multimedia, vol. 21, no. 1, pp. 234–245 (2019) doi: 10.1109/tmm.2018.2856094
3. Bautista-Garcia, M., Feria-Revano-Jr, T., Cunanan-Yabut, A.: Hand alphabet recognition for dactylology conversion to english print using streaming video segmentation. pp. 46–51 (2021) doi: 10.1145/3479162.3479169
4. Bellen, E. M., Mendoza, J. R. M., Seroy, D. G. T., Ong, D., de Guzman, J. A.: Integrated visual-based ASL captioning in videoconferencing using CNN. pp. 1–6 (2022) doi: 10.1109/TENCON55691.2022.9977526
5. Bennett-Gayle, D., Yuan, X., Knight, T.: The coronavirus pandemic: Accessible technology for education, employment, and livelihoods. Assistive Technology, pp. 1–8 (2021) doi: 10.1080/10400435.2021.1980836
6. Boulares, M., Jemni, M.: Methodological foundation for sign language 3D motion trajectory analysis. Lecture Notes in Computer Science, vol. 7619, pp. 67–77 (2012) doi: 10.1007/978-3-642-34156-4_8
7. Boulares, M., Jemni, M.: Automatic hand motion analysis for the sign language space management. Pattern Analysis and Applications, vol. 22, no. 2, pp. 311–341 (2019) doi: 10.1007/s10044-017-0631-x
8. Bragg, D., Koller, O., Bellard, M., Berke, L., Boudreault, P., Braffort, A., Caselli, N., Huenerfauth, M., Kacorri, H., Verhoef, T., Vogler, C., Ringel-Morris, M.: Sign language recognition, generation, and translation: An interdisciplinary perspective. In: The 21st International ACM SIGACCESS Conference on Computers and Accessibility, pp. 16–31 (2019) doi: 10.1145/3308561.3353774
9. Chen, Z., Luo, Y., Mesgarani, N.: Deep attractor network for single-microphone speaker separation. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 246–250 (2017) doi: 10.1109/icassp.2017.7952155
10. Chuan, C. H., Guardino, C. A.: Designing smartsignplay: An interactive and intelligent american sign language app for children who are deaf or hard of hearing and their families. In: Companion Publication of the 21st International Conference on Intelligent User Interfaces, pp. 45–48 (2016) doi: 10.1145/2876456.2879483

11. Cook, A. M., Miller-Polgar, J.: Cook and Hussey's assistive technologies: Principles and practice. Elsevier Health Sciences (2007)
12. Di-Gregorio, M., Sebillo, M., Vitiello, G., Pizza, A., Vitale, F.: Prosign everywhere - addressing communication empowerment goals for deaf people. In: Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good, pp. 207–212 (2019) doi: 10.1145/3342428.3342695
13. Faltaous, S., Winkler, T., Schneegass, C., Gruenefeld, U., Schneegass, S.: Understanding challenges and opportunities of technology-supported sign language learning. In: ACM International Conference Proceeding Series, pp. 15–25 (2022) doi: 10.1145/3519391.3519396
14. Fang, B., Co, J., Zhang, M.: DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems, pp. 1–13 (2017) doi: 10.1145/3131672.3131693
15. Glasser, A., Mande, V., Huenerfauth, M.: Understanding deaf and hard-of-hearing users' interest in sign-language interaction with personal-assistant devices. In: Proceedings of the 18th International Web for All Conference, pp. 1–11 (2021) doi: 10.1145/3430263.3452428
16. Glasser, A., Watkins, M., Hart, K., Lee, S., Huenerfauth, M.: Analyzing deaf and hard-of-hearing users' behavior, usage, and interaction with a personal assistant device that understands sign-language input. In: Proceedings of the Conference on Human Factors in Computing Systems, pp. 1–12 (2022) doi: 10.1145/3491102.3501987
17. Gupta, K., Singh, A., Yeduri, S. R., Srinivas, M. B., Cenkeramaddi, L. R.: Hand gestures recognition using edge computing system based on vision transformer and lightweight CNN. Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 3, pp. 2601–2615 (2023) doi: 10.1007/s12652-022-04506-4
18. Hassan, S., Amin, A. A., Gordon, A., Lee, S., Huenerfauth, M.: Design and evaluation of hybrid search for american sign language to English dictionaries: Making the most of imperfect sign recognition. pp. 1–13 (2022) doi: 10.1145/3491102.3501986
19. Huenerfauth, M., Gale, E., Penly, B., Pillutla, S., Willard, M., Hariharan, D.: Evaluation of language feedback methods for student videos of american sign language. ACM Transactions on Accessible Computing, vol. 10, no. 1, pp. 1–30 (2017) doi: 10.1145/3046788
20. Huenerfauth, M., Gale, E., Penly, B., Willard, M., Hariharan, D.: Comparing methods of displaying language feedback for student videos of american sign language. In: Proceedings of the 17th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 139–146 (2015) doi: 10.1145/2700648.2809859
21. Huenerfauth, M., Lu, P.: Effect of spatial reference and verb inflection on the usability of sign language animations. Universal Access in the Information Society, vol. 11, no. 2, pp. 169–184 (2011) doi: 10.1007/s10209-011-0247-7
22. Irving, A., Foulds, R.: A parametric approach to sign language synthesis. In: Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 212–213 (2005) doi: 10.1145/1090785.1090835
23. Jiménez-Salas, J., Chacón-Rivas, M.: A systematic mapping of computer vision-based sign language recognition. In: International Conference on Inclusive Technologies and Education, pp. 1–11 (2022) doi: 10.1109/CONTIE56301.2022.10004413
24. Johnson, R.: Towards enhanced visual clarity of sign language avatars through recreation of fine facial detail. Machine Translation, vol. 35, no. 3, pp. 431–445 (2021) doi: 10.1007/s10590-021-09269-x
25. Jones, M., Bench, N., Ferons, S.: Vocabulary acquisition for deaf readers using augmented technology. In: 2nd Workshop on Virtual and Augmented Assistive Technology, pp. 13–15 (2014) doi: 10.1109/VAAT.2014.6799461

26. Kacorri, H., Harper, A., Huenerfauth, M.: Comparing native signers' perception of american sign language animations and videos via eye tracking. In: Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 1–8 (2013) doi: 10.1145/2513383.2513441
27. Kacorri, H., Harper, A., Huenerfauth, M.: Measuring the perception of facial expressions in american sign language animations with eye tracking. Universal Access in Human-Computer Interaction Design for All and Accessibility Practice, pp. 553–563 (2014) doi: 10.1007/978-3-319-07509-9_52
28. Kacorri, H., Huenerfauth, M.: Comparison of finite-repertoire and data-driven facial expressions for sign language avatars. Lecture Notes in Computer Science, vol. 9176, pp. 393–403 (2015) doi: 10.1007/978-3-319-20681-3_37
29. Kacorri, H., Huenerfauth, M.: Continuous profile models in ASL syntactic facial expression synthesis. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 2084–2093 (2016) doi: 10.18653/v1/p16-1196
30. Kitchenham, B.: Procedures for performing systematic reviews. Keele University, vol. 33, pp. 1–26 (2004)
31. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering (2007)
32. Kudrinko, K., Flavin, E., Shepertycky, M., Li, Q.: Assessing the need for a wearable sign language recognition device for deaf individuals: Results from a national questionnaire. Assistive Technology, vol. 34, no. 6, pp. 684–697 (2022) doi: 10.1080/10400435.2021.1913259
33. Kurtoglu, E., Gurbuz, A. C., Malaia, E., Griffin, D., Crawford, C., Gurbuz, S. Z.: Sequential classification of ASL signs in the context of daily living using RF sensing. In: IEEE National Radar Conference, pp. 1–6 (2021) doi: 10.1109/RadarConf2147009.2021.9455178
34. Kurtoglu, E., Gurbuz, A. C., Malaia, E. A., Griffin, D., Crawford, C., Gurbuz, S. Z.: ASL trigger recognition in mixed activity/signing sequences for RF sensor-based user interfaces. IEEE Transactions on Human-Machine Systems, vol. 52, no. 4, pp. 699–712 (2022) doi: 10.1109/thms.2021.3131675
35. Labonnote, N., Høyland, K.: Smart home technologies that support independent living: challenges and opportunities for the building industry – a systematic mapping study. Intelligent Buildings International, vol. 9, no. 1, pp. 40–63 (2015) doi: 10.1080/17508975.2015.1048767
36. Lee, B. G., Lee, S. M.: Smart wearable hand device for sign language interpretation system with sensors fusion. IEEE Sensors Journal, vol. 18, no. 3, pp. 1224–1232 (2018) doi: 10.1109/jsen.2017.2779466
37. Lee, M., Bae, J.: Real-time gesture recognition in the view of repeating characteristics of sign languages. IEEE Transactions on Industrial Informatics, vol. 18, no. 12, pp. 8818–8828 (2022) doi: 10.1109/TII.2022.3152214
38. Lu, P., Huenerfauth, M.: Accessible motion-capture glove calibration protocol for recording sign language data from deaf subjects. In: Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 83–90 (2009) doi: 10.1145/1639642.1639658
39. Lu, P., Huenerfauth, M.: Data-driven synthesis of spatially inflected verbs for american sign language animation. ACM Transactions on Accessible Computing, vol. 4, no. 1, pp. 1–29 (2011) doi: 10.1145/2039339.2039343
40. Lu, P., Huenerfauth, M.: Collecting and evaluating the CUNY ASL corpus for research on american sign language animation. Computer Speech and Language, vol. 28, no. 3, pp. 812–831 (2014) doi: 10.1016/j.csl.2013.10.004
41. Lucas, C., Bayley, R., Valli, C.: What's your sign for PIZZA? An introduction to variation in American Sign Language. Gallaudet University Press (2003)

42. Mande, V., Glasser, A., Dingman, B., Huenerfauth, M.: Deaf users' preferences amongwake-up approaches during sign-language interaction with personal assistant devices. In: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, pp. 1–6 (2021) doi: 10.1145/3411763.3451592
43. Mande, V., Glasser, A., Dingman, B., Huenerfauth, M.: Deaf users' preferences among wake-up approaches during sign-language interaction with personal assistant devices. In: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, pp. 1–6 (2021) doi: 10.1145/3411763.3451592
44. Mathew, R., Mak, B., Dannels, W.: Access on demand: Real-time, multi-modal accessibility for the deaf and hard-of-hearing based on augmented reality. In: Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 1–6 (2022) doi: 10.1145/3517428.3551352
45. Myers, M. J., Annis, I. E., Withers, J., Williamson, L., Thomas, K. C.: Access to effective communication aids and services among american sign language users across North Carolina: Disparities and strategies to address them. Health Communication, vol. 37, no. 8, pp. 962–971 (2021) doi: 10.1080/10410236.2021.1878594
46. Naranjo-Zeledón, L., Peral, J., Ferrández, A., Chacón-Rivas, M.: A systematic mapping of translation-enabling technologies for sign languages. Electronics, vol. 8, no. 9 (2019) doi: 10.3390/electronics8091047
47. National Institute on Deafness and Other Communication Disorders: Assistive devices for people with hearing, voice, speech, or language disorders (2019) www.nidcd.nih.gov/health /assistive-devices-people-hearing-voice-speech-or-language-disorders
48. Neidle, C., Kegl, J., MacLaughlin, D., Bahan, B., Lee, R. G.: The syntax of american sign language: Functional categories and hierarchical structure. The MIT Press (2000)
49. Othman, A., Hamdoun, R.: Toward a new transcription model in XML for sign language processing based on gloss annotation system. In: 4th International Conference on Information and Communication Technology and Accessibility, pp. 1–5 (2013) doi: 10.1 109/ICTA.2013.6815317
50. Pannattee, P., Kumwilaisak, W., Hansakunbuntheung, C., Thatphithakkul, N.: Novel american sign language fingerspelling recognition in the wild with weakly supervised learning and feature embedding. In: 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 291–294 (2021) doi: 10.1109/ECTI-CON51831.2021.9454677
51. Paudyal, P., Banerjee, A., Gupta, S. K. S.: SCEPTRE: A pervasive, non-invasive, and programmable gesture recognition technology. In: Proceedings of the 21st International Conference on Intelligent User Interfaces, pp. 282–293 (2016) doi: 10.1145/2856767.28 56794
52. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, vol. 64, pp. 1–18 (2015) doi: 10.1016/j.infsof.2015.03.007
53. Phadtare, L. K., Kushalnagar, R. S., Cahill, N. D.: Detecting hand-palm orientation and hand shapes for sign language gesture recognition using 3D images. In: Western New York Image Processing Workshop, pp. 29–32 (2012) doi: 10.1109/WNYIPW.2012.6466652
54. Rahman, M. M., Kurtoglu, E., Mdrafi, R., Gurbuz, A. C., Malaia, E., Crawford, C., Griffin, D., Gurbuz, S. Z.: Word-level ASL recognition and trigger sign detection with RF sensors. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8233–8237 (2021) doi: 10.1109/icassp39728.2021.9414063
55. Rui-Xia-Ang, J., Liu, P., McDonnell, E., Coppola, S.: In this online environment, we're limited: Exploring inclusive video conferencing design for signers. In: CHI Conference on Human Factors in Computing Systems, pp. 1–16 (2022) doi: 10.1145/3491102.3517488

56. Schnepp, J. C., Wolfe, R. J., McDonald, J. C., Toro, J. A.: Combining emotion and facial nonmanual signals in synthesized american sign language. In: Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility, pp. 249–250 (2012) doi: 10.1145/2384916.2384977
57. Sriram, N., Nithiyanandham, M.: A hand gesture recognition based communication system for silent speakers. International Conference on Human Computer Interactions, pp. 1–5 (2013) doi: 10.1109/ICHCI-IEEE.2013.6887815
58. The Indiana University: Knowledge base, types of impairments (2019)
59. Weaver, K. A., Starner, T., Hamilton, H.: An evaluation of video intelligibility for novice american sign language learners on a mobile device. In: Proceedings of the 12th International ACM SIGACCESS Conference on Computers and accessibility, pp. 107–114 (2010) doi: 10.1145/1878803.1878824
60. Wolfe, R., McDonald, J., Toro, J., Baowidan, S., Moncrief, R., Schnepp, J.: Promoting better deaf/hearing communication through an improved interaction design for fingerspelling practice. Universal Access in Human-Computer Interaction, pp. 495–505 (2015) doi: 10.100 7/978-3-319-20678-3_48
61. World Health Organization: Assistive technology (2023) www.who.int/news-room/fact-she ets/detail/assistive-technology
62. Yeratziotis, A., Achilleos, A., Koumou, S., Thibodeau, R. A., Vanezi, E., Geratziotis, G., Papadopoulos, G. A., Iasonas, I., Yeratziotis, A.: Accessible system and social media mobile application for deaf users: ASM4Deaf. In: Proceedings of the ACM Conference on Information Technology for Social Good, pp. 39–47 (2022) doi: 10.1145/3524458.3547234

ISSN 1870-4069

Invertible Neural Networks for Inference Integrity Verification

Malgorzata Schwab, Ashis Biswas

[1] University of Colorado, Denver,
USA

{malgorzata.schwab,ashis.biswas}@ucdenver.edu

Abstract. In this study we explore the topic of Trustworthy AI and how reversibility in neural networks can play a role in protecting machine learning applications. We propose a framework to enhance machine learning systems robustness through the integrity verification across the inference pipeline of a deployed model and apply a concept of a Trusted Neural Network, which provides a system engineering abstraction to implement it. We leverage the Invertible Neural Network architecture with its remarkable data reconstruction and anomaly detection capabilities to validate that the inference flow pipeline is intact and thus the network prediction can be trusted, as trained. The result of that assessment is measured as an Inference Integrity Score and can be reported in real time to safeguard system integrity and suppress suspicious results. We propose an AI firewall in the form of test nodes implementing the Trusted Neural Network interface comprising an input verification layer in front of the running models, participating in the workflow. This easy to implement verification-based paradigm offers a pragmatic approach to achieve machine learning robustness and takes a step towards Trustworthy AI.

Keywords: Integrity, invertible, reversibility, robustness, trustworthiness.

1 Introduction

With the explosion of AI-augmented systems that are impacting millions around the world each day, we need to ensure that those systems are trustworthy, robust, and protected to stand up against adversarial attacks. This concept paper is inspired by the increasing role of machine learning in the decision-making process across a wide spectrum of domains, which brings to the forefront the importance of verifying the integrity of end-to-end inference flow, so the outcome of the system can be trusted.

We propose that the general solution architecture paradigm for any mission critical decision support system that leverages machine learning components incorporates a layer of integrity verification around a running model to ensure trustworthiness of the pipeline. Our technique is applicable to machine learning inference flows significant enough to be protected by an extra security layer.

We build upon the concept of a Trusted Neural Network (TNN) [1], which leverages Invertible Neural Network (INN) architecture based on a revolutionary approach to achieve reversibility in neural networks introduced by Dinh [2] and subsequently incorporated into a framework by Ardizone [3]. An INN, which is invertible by

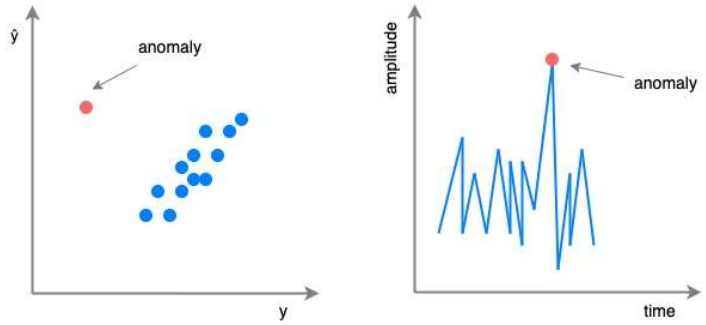


Fig. 1. Anomalies visualized [5].

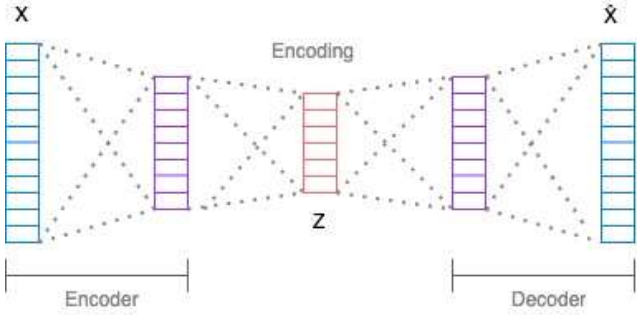


Fig. 2. Classic autoencoder [8].

construction, offers a remarkable data reconstruction capability that can be leveraged to validate that the inference flow pipeline is intact and that the output of it can be trusted.

The result of that assessment, which we call the Inference Integrity Score, can be reported in real time and acted upon to safeguard system integrity by suppressing suspicious outcomes. The implementation of our Trustworthy AI paradigm employs the TNN-based test nodes comprising an AI-firewall layer offers a pragmatic approach to protecting machine learning pipelines and does not require any intricate intervention into the models themselves to handle adversarial inputs.

The remainder of this paper is organized as follows: Section 2 briefly reviews related work pertaining to safeguarding machine learning inference pipelines. It then elaborates on anomaly detection techniques [8] and Invertible Neural Networks touching upon normalizing flows [2] - the theory underlying the reversibility of deep neural networks. We also introduce the Framework for Easily Invertible Architectures (FrEIA) previously established by Ardizone [3], which provides an SDK to construct custom INN configurations to make it quick and approachable.

We then discuss the remarkable ability of an Invertible Neural Network to reconstruct data from its compressed latent representation, outperforming traditional autoencoder architecture. In Section 3 we look at the Trusted Neural Network API and

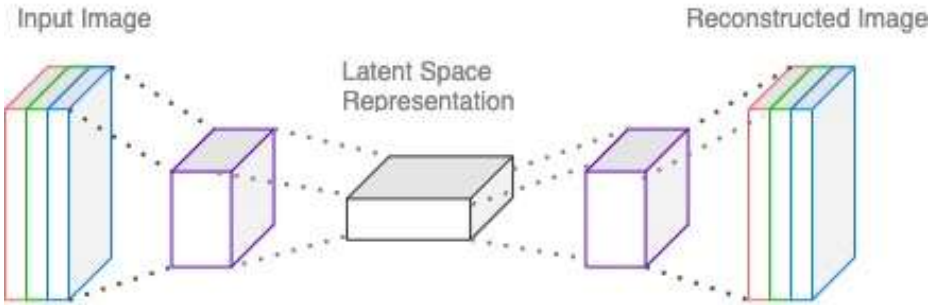


Fig. 3. Convolutional Autoencoder [8].

learn how a TNN node can be incorporated into a verification-based inference protection layer. Section 4 summaries the study and offers conclusion.

2 Related Works

Machine learning system robustness, which encompasses building reliable, resilient, and fault-tolerant machine learning systems, is an active area of research. Much attention is given to strengthen adversarial resistance of the deep learning models themselves, but a test-based verification-driven approach to validate the inference pipeline provides an effective scheme to improve system robustness, while narrowing the gap between machine learning research and practice. The risks related to the inference pipeline's state of integrity can be effectively mitigated by verifying the reasonability of the prediction outcome, which is discussed by Apruzzese in the methodology survey "Real Attackers Don't Compute Gradients" [4].

2.1 Anomaly Detection

We invoke the topic of anomaly detection as relevant to the verification of the inference pipeline integrity. Anomaly detection is a process of identifying data that does not fit into a pattern of what is expected.

As described in [5] and depicted in Figure 1, abnormal patterns in the phenomena characterized by low dimensionality can be easily discovered with an algorithmic approach based on acceptable value ranges, with simple clustering techniques, or even assessed visually.

Giannoni [5] and subsequently Yin [6] put anomaly detection methods in several categories, such as statistical-based methods, probability-based methods, similarity-based methods, and most recent prediction-based methods. The high dimensional scenarios surrounding systems with machine learning components highly dependent on integrity of the data, however, require more sophisticated multivariate statistics methods based on probability distributions and deep learning techniques.

They are exemplified by generative neural networks, such as several classes of autoencoders, including novel INN-based autoencoders described [8] based on Invertible Neural Networks trained for anomaly detection.

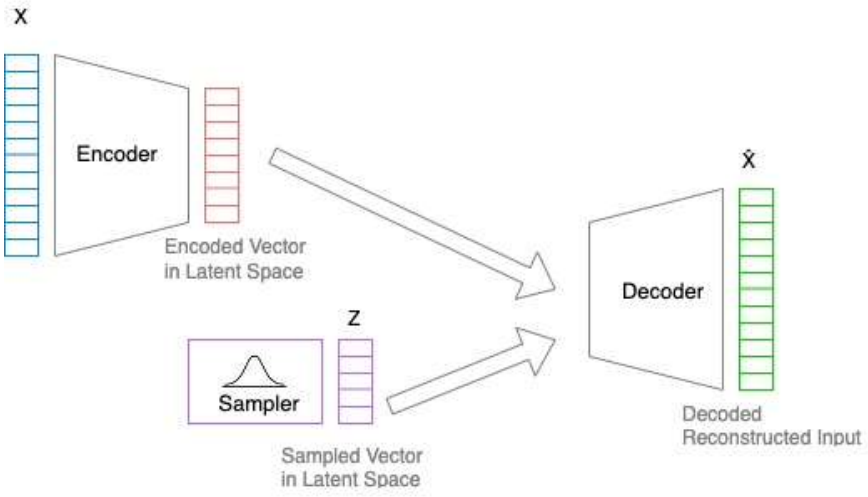


Fig. 4. Variational Autoencoder [8].

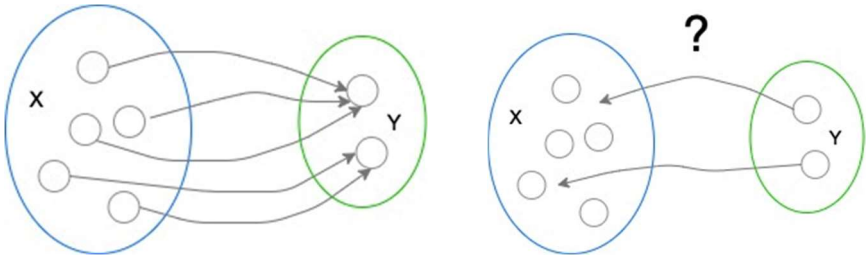


Fig. 5. Forward mapping of x → y (left) and Inverse ambiguity (right).

2.2 Autoencoders

Autoencoders belong to the family of unsupervised deep learning neural network models well suited for dimensionality reduction and have been described extensively in numerous works, such as [5] and [6], then referenced in [8]. The general idea around this type of neural network is to extract the most relevant features from input data and then learn how to reconstruct the original data from its compressed representation.

For unexpected inputs, which the model has not seen during training, the reconstruction error should be higher, and crossing a configurable threshold, dependent on a problem domain, constitutes an anomaly. As described in [8] and shown in Fig. 2, a classic autoencoder consists of an encoder and a decoder, implemented as fully connected neural networks.

The encoder compresses the network input x into a lower dimensional latent representation z defined by the bottleneck. The decoder takes the output of the encoder

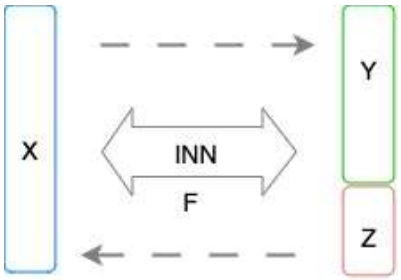


Fig. 6. Invertible Neural Network Conceptual Diagram.

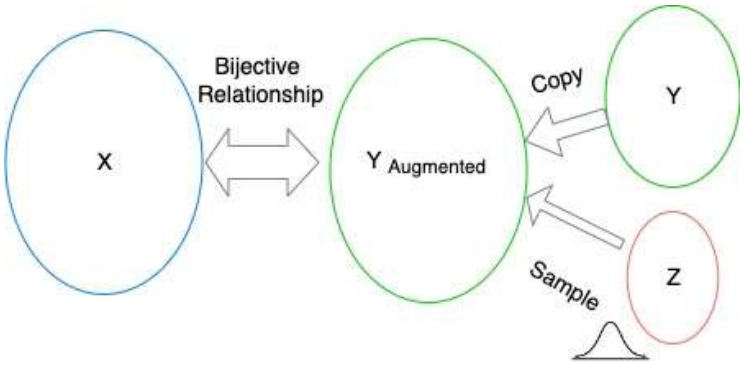


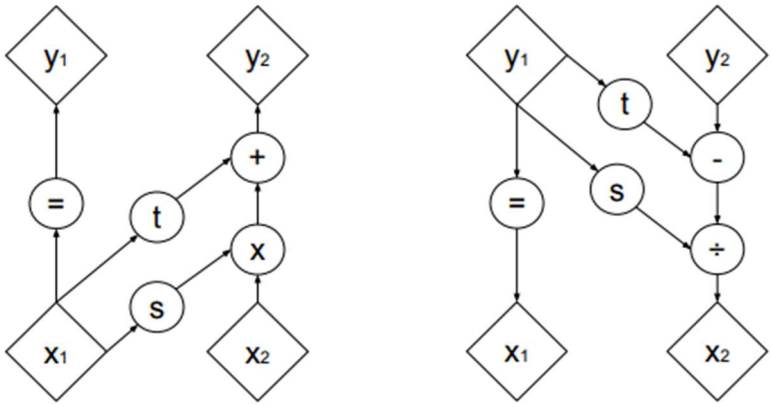
Fig. 7. Reconstructing phenomenon X from observation Y.

and decodes the latent representation back to the original input $\hat{x}$. The information preserved in hidden neurons is considered as the encoded features. The learning process is based on minimizing the reconstruction error, which is assessed by comparing the reconstructed input with the original one. The learned representation corresponds to the final hidden state of the encoder network and acts like a summary of the input sequence.

There are several variations of autoencoder architecture [8], such as a convolutional autoencoder, depicted in Fig. 3, which uses convolutional layers to create a compressed representation [6], or a variational autoencoder depicted in Fig. 4, capable not only of reconstructing the original input, but also enhancing it by generating new content based on the sampling from the learned probability density distribution of the input domain.

A compress-reconstruct type of a challenge reflected in the autoencoder encoder-decoder architecture belongs to the class of "ill-posed" inverse problems, which are characterized by inherent ambiguity due to the existence of an information bottleneck. Such problems have been successfully addressed by the reversible neural network architecture applied in Invertible Neural Networks, which makes them an interesting option to help with our integrity verification undertaking.

In this work we leverage previous findings and principles regarding several types of autoencoders together with reversible neural networks and apply the INN-based architecture for anomaly detection as a core of the TNN network integrity verification nodes.



(a) Forward propagation (b) Inverse propagation

Fig. 8. Real NVP Affine Coupling Block [2].

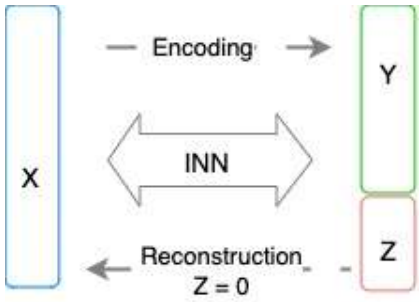


Fig. 9. INN as Autoencoder [8].

2.3 Invertible Neural Network

As explored in [8] and referenced here for context, an Invertible Neural Network is a class of networks suited to solve ambiguity that characterizes inverse problems, where multiple parameter sets can produce the same observed outcome, as depicted in Fig. 5. To express this ambiguity, the posterior probability of the parameters' distribution, given an outcome y, must be learned so the most appropriate set can be selected.

Such a model can perform log-density estimation of data points, leading to efficient inference and precise reconstruction of the inputs from the hierarchical features extracted by the model. This extraordinary capability to reconstruct the inputs corresponding to the encoder-decoder functionality makes INN a natural candidate to help solve the problem of anomaly detection.

An INN is trained simultaneously in the forward and reverse directions, Fig. 6. The forward learning process uses additional latent output variables to capture information otherwise lost, making the learning of the inverse process explicit. To solve the general inverse problem, we augment the observation space Y with a latent variable Z which follows a normal distribution and look for a bijective function F that can map Z back to $\hat{X}$.

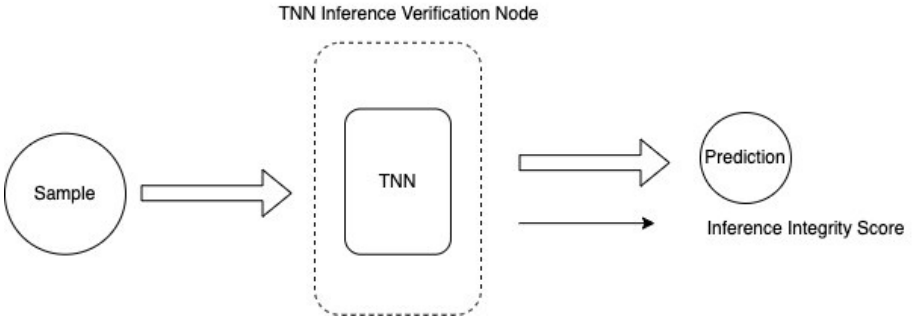


Fig. 10. TNN Context Diagram [1].

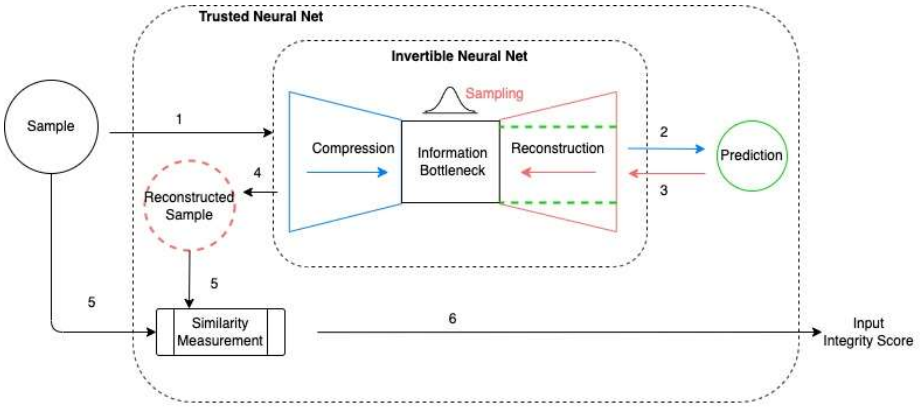


Fig. 11. TNN Architecture.

An INN learns an invertible, stable, mapping between a data distribution PX and a latent distribution PZ, typically Gaussian, as shown in Fig. 7. Invertibility of neural networks was spearheaded by Dinh [2] as "real-valued non-volume preserving transformations" (Real NVP) architecture, who introduced a stack of invertible affine coupling blocks (Fig. 8), arranged in hidden layers. Given a D-dimensional input x and d < D, the output y of an affine coupling layer follows the following equations [2]:

$$y_{1:d} = x_{1:d}, \tag{1}$$

$$y_{d+1:D} = x_{d+1:D} \odot \exp[s(x_{1:d}) + t(x_{1:d})], \tag{2}$$

where s and t are functions from $R^d \rightarrow R^{D-d}$, and $\odot$ is the Hadamard product or element-wise product. Each block splits its input and output into two parts and applies transformations s (scale) and t (translation), which themselves do not have to be invertible – they can be quite complex and are often implemented as artificial neural networks, such as a CNNs.

It has been proven [3] that a stack of such invertible blocks makes the end-to-end layout also invertible. Based on this architecture, the Invertible Neural Network guarantees reversibility by its construction and solves the ambiguous inverse relationships directly.

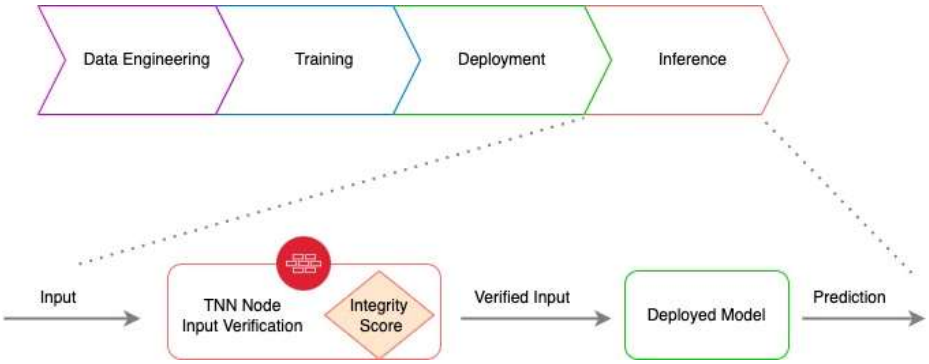


Fig. 12. A TNN node for input integrity verification.

2.4 INN Trained as an Autoencoder

As demonstrated by Nguyen [7] on MNIST, CIFAR and CelebA, and recently by Schwab [8] for time series data, an INN has superb capability for anomaly detection. It compared an INN-based implementation to conventional autoencoders for different bottleneck sizes, which demonstrated that INN autoencoders can achieve similar or better reconstruction results.

It showed that the architecture restrictions on INN autoencoders to ensure invertibility do not negatively affect their performance, while the advantages of INNs are still preserved. This entails a tractable Jacobian for both forward and inverse mapping as well as explicit computation of posterior probabilities.

It also provided an explanation for the saturation in reconstruction loss for large bottleneck sizes in classical autoencoders and concluded that an INN might not have any intrinsic information loss and thereby are not constrained by a maximal depth after which only suboptimal results can be achieved.

The concept of an INN entails bijective input-output mapping, so the dimensions of input x and output y augmented with z must be equal. As depicted in Fig. 9 below, an artificial bottleneck must be constructed to achieve autoencoder-like behavior. It is accomplished by zeroing the latent **z** to make sure that no extra information is retained by the network in the inverse process of representation learning.

As demonstrated in [8], the reconstruction loss on the anomalous samples across a variety of datasets was an order of magnitude greater as compared to the reconstruction error on the healthy validation data. The INN-autoencoder architecture also shows excellent performance, which renders it as an effective tool for the inference integrity verification task.

2.5 Trusted Neural Network

The diagram in Fig. 10 below depicts a conceptual template of a system comprising a Trusted Neural Network conceptualized in [1], where the output, in addition to the predicted result, includes an Inference Integrity Score to help assess trustworthiness of the outcome.

```
Request:

url = 'http://api.tnn.com/'
params = {'query': 'node_1'}
response = requests.get (url, params)
response.json()
```

```
Response:

Output:
{'confidence': 0.777,
 'prediction': 'compromized',
 'Inference Integrity Score': 0.987}
```

Fig. 13. TNN API Request and Response.

It leverages the capability of an Invertible Neural Network deal with inverse problems and to reconstruct an input from an output, in their respective domains.TNN is a general solution architecture paradigm and the concrete implementations reflecting the needs of specific problem domains can be derived from there. Current methodologies employed to verify the integrity of Artificial Neural Networks leverage sampling strategies, which operate in the outer perimeter of the network.

The TNN concept, however, incorporates the integrity measure as an integral part of the system. We propose that the inference flow is augmented with the inverse output-to-input verification steps, and that the INN-based Trusted Neural Network stackable nodes assume this responsibility – trained on the respective datasets, they are tasked with detecting and suppressing suspicious out-of-distribution data anomalies along the pipeline.

3 Proposed Framework for Inference Integrity Verification

3.1 Trusted Neural Network Architecture

A TNN (Fig. 11) used as the module integrity verification node is composed of several high-level building blocks, each of which is independently defined, can be independently improved, and empirically tuned to fit the needs of any individual application use case.

The integrity measure is computed by comparing an original input sample with the sample reconstructed by the Invertible Neural Network component embedded inside the TNN, and if too low, the overall prediction shall be discarded.

3.2 Information Bottleneck Principle

The INN is optimized along the principles of the Information Bottleneck Theory [10, 11] (alluded to in Fig. 11), capable of balancing the purposeful information loss against the desired accuracy of the model. The Information Bottleneck method measures how well Y can be predicted from a compressed representation Z, compared to its direct prediction from X. The algorithm minimizes the loss function L with respect to conditional distribution $p(z|x)$:

$$L_{IB} = I(X,Z) - \beta\, I\,(Y,Z), \tag{3}$$

where $I(X;Z)$ and $I(Z;Y)$ are the mutual information of X and Z, and Y and Z respectively, and β is Lagrange multiplier.

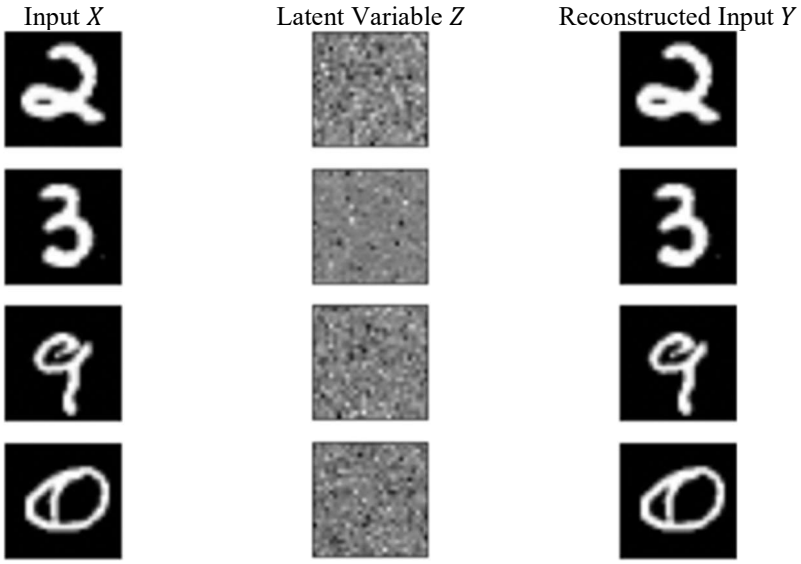


Fig. 14. MNIST Experiment.

3.3 Trustworthy AI Solution Architecture

We propose a novel type of test-driven approach to ensure ML integrity, depicted in Fig. 12, which leverages the TNN nodes to protect against adversarial data at any given step of the inference pipeline, and thus guarding its integrity. The solution employs one or more Trusted Neural Network node(s) with INN at its heart configured for data reconstruction, so that the inputs of the modules comprising a pipeline can be subjected to a test, as indicated in Fig. 12 steps 1-6.

Input and outputs of a module may or may not be in the data domain, which is the strength of Invertible Neural Networks, as compared to the classic autoencoder architecture. The similarity measure and the thresholds would vary per use case, and thus they must be designed specifically for any given domain:

$$\left\| X_{\text{inverted}} - X_{\text{original}} \right\| < \text{Reconstrucion Error Margin.} \tag{4}$$

The Trusted Neural Network design pattern comes with REST API [12], depicted in Fig. 13, which in addition to the prediction outcome also returns the Inference Integrity Score. The proposed standard would add the Integrity Score parameter to the ML API response payload as an integrated workflow security measure.

3.4 Input Reconstruction

Several experiments were conducted to verify various INN configurations with respect to reconstructing the most probable input given an output. Described in [1], they followed the implementation examples provided in [3] using synthetic points data sets.

Another experimental INN, configured to process the MNIST data set, tested successfully as well (Fig. 14). The forward pass through the invertible network gives us a latent image Z, which fed to the network in the reversed flow outputs a regenerated X, noted as X_{inverted}:

$$Z = INN_{\text{forward}}(X_{\text{original}}), \tag{5}$$

$$X_{\text{inverted}} = INN_{\text{reverse}}(Z). \tag{6}$$

The difference between the original input X entering the TNN and its counterpart X_{inverted} regenerated by the network in the reverse flow is negligible:

$$\left\| X_{\text{inverted}} - X_{\text{original}} \right\| < 1e-5. \tag{7}$$

A result like that which would be reflected in a high value of Inference Integrity Score and provide a successful test for a TNN node at a given step of the inference flow.

4 Summary and Conclusion

This work proposes an easy to implement pragmatic scheme to enhance robustness of machine learning systems through a test-driven inference flow verification layer based on the Trusted Neural Network nodes and their API abstraction. It leverages the Invertible Neural Network architecture and an open-source framework to construct the INN-based state-of-the-art anomaly detector.

The paradigm is generalizable across problem domains and aspires to become a useful practice in drafting robust high-level solution architectures for systems which incorporate machine learning capabilities and can benefit from additional measures of trustworthiness.

References

1. Schwab, M., Kumer-Biswas, A.: Trusted neural network (TNN); Reversibility in neural networks for inference integrity verification. In: Proceedings of the International Conference on Machine Learning and Applications (2023)
2. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real NVP. In: Proceedings of the International Conference on Learning Representations (2016) doi: 10.48550/ARXIV.1605.08803
3. Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., Köthe, U.: Analyzing inverse problems with invertible neural networks. In: Proceedings of the International Conference on Learning Representations (2019) doi: 10.48550/ARXIV.1808.04730
4. Apruzzese, G., Anderson, H. S., Dambra, S., Freeman, D., Pierazzi, F., Roundy, K. A.: Real attackers don't compute gradients: Bridging the gap between adversarial ML research and practice. In: IEEE Conference on Secure and Trustworthy Machine Learning (2022) doi: 10.48550/ARXIV.2212.14315
5. Giannoni, F., Mancini, M., Marinelli, F.: Anomaly detection models for IoT time series data (2018) doi: 10.48550/ARXIV.1812.00890

6. Yin, C., Zhang, S., Wang, J., Xiong, N. N.: Anomaly detection based on convolutional recurrent autoencoder for IoT time series. IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 1, pp. 112–122 (2022) doi: 10.1109/tsmc.2020.2968516
7. Nguyen, T. L., Ardizzone, L., Köthe, U.: Training invertible neural networks as autoencoders. In: German Conference on Pattern Recognition, vol. 11824, pp. 442–455 (2019) doi: 10.1007/978-3-030-33676-9_31
8. Schwab, M., Biswas, A.: Invertible neural network for time series anomaly detection. In: 8th International Conference on Software Engineering, pp. 227–239 (2023) doi: 10.5121/csit.2023.131220
9. Sutskever, I., Vinyals, O., Le, Q. V.: Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 2, pp. 3104--3112 (2014)
10. Tishby, N., Zaslavsky, N.: Deep learning and the information bottleneck principle. In: IEEE Information Theory Workshop (2015) doi: 10.48550/ARXIV.1503.0240
11. Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., Cox, D. D.: On the information bottleneck theory of deep learning. In: International Conference on Learning Representations (2018)
12. Fielding, T.: Architectural styles and the design of network-based software architectures. Dissertation submitted in partial satisfaction of the requirements for the degree of doctor of philosophy, University of California (2000) www.ics.uci.edu/~fielding/pubs/dissertation/fieldi ng_dissertation.pdf

ISSN 1870-4069

Hand Gesture Recognition Applied to the Interaction with 3D Models in Virtual Reality

Ángel Leonardo Valdivieso-Caraguay, Óscar Mauricio Rivera-Cajía, Bryan Norberto Flores-Sarango, Lorena Isabel Barona-López, Marco E. Benalcázar

Escuela Politécnica Nacional,
Departamento de Informática y Ciencias de la Computación,
Artificial Intelligence and Computer Vision Research Lab,
Ecuador

angel.valdivieso@epn.edu.ec

Abstract. In this work, we present a real-time hand gesture based Human Computer Interaction (HCI) system for control a Virtual Reality (VR) application by using of Oculus Rift and Myo armband. For this purpose, an Hand Gesture Recognition (HGR) model and a VR application were implemented. The K-Nearest Neighbors (KNN) and Dynamic Time Warping (DTW) algorithms were applied to develop the HGR model. The inputs to this model are signals of 11 hand gestures measured by Myo Armband and G-Force Pro using their built-in surface electromyography (EMG) dry sensors and inertial measurement unit (IMU). The outcome of the HGR model is the designation that characterizes the gesture performed by the user. The VR application was developed by the game engine Unity using Oculus Rift as input device into virtual environment. It allows navigate over an interface and manipulate three-dimensional (3D) objects taking advantage of their properties for a sophisticated experience. The HGR model is used in the VR application where each identified gesture performs an action. The system present a natural communication through hand gestures in a virtual environment. In average, we achieved real-time gesture classification with an accuracy of 82% on eleven distinct gestures. The SUS test results rank our system as excellent in terms of usability.

Keywords: Human computer interaction, hand gesture recognition, virtual reality, K-nearest neighbors, dynamic time warping, electromyography, inertial measurement unit.

1 Introduction

The study of interfaces between humans and computers is known as Human Computer Interaction (HCI). Traditional HCI methods such as keyboards, mouse or touch screens are often unfriendly when interacting with computers. The study of the field of gesture recognition in combination with HCI transcended this barrier because the use of gestures is a more natural way to provide an interface between a user and a computer [30, 1]. These improvements in HCI technology are also leading to advances in another field closely related to HCI: virtual reality (VR).

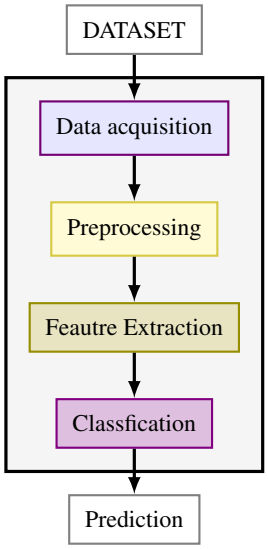


Fig. 1. Hand gesture system framework.

Hand Gesture Recognition (HGR) models are human–computer systems that determine what gesture was performed and when it is performed [12]. In this work, a gesture recognition system based on EMG and IMU data is divided into 4 stages: data acquisition, preprocessing, feature extraction and classification [7] as shown in Figure 1. Those models can acquire data with different instruments, such as gloves, vision sensors, inertial measurement units (IMUs), surface electromyography sensors or a combinations of devices [12].

In this work we combine surface EMG and IMU sensors. Surface EMG is a technique that records the action potentials of the muscle fibers with surface sensors [14]. IMU is a device that records four features: velocity, shape, location and orientation in motion capture on specific body by using a combination of accelerometer, gyroscopy and magnetometer sensor [19, 5]. The acquired information is combined depending on the two categories of gestures according to their type of interaction [27]:

– Static gestures - gestures based on a single posture that is maintained for a certain amount of time.

– Dynamic gestures - gestures based on a motion trajectory.

On this context, we propose a HGR model based on static and dynamic gestures measured by the dry surface EMG and IMU sensors built-in the Myo Armband. Additionally, we propose a virtual reality application controlled by hand gestures where our HGR model can be used to control it.

The application provides a natural interaction of actions through gestures [9, 23]. In addition, it allows a detailed examination of the 3D models. Virtual reality gives full control of the environment to the user through the glasses movement and, with our proposal, hand gestures.

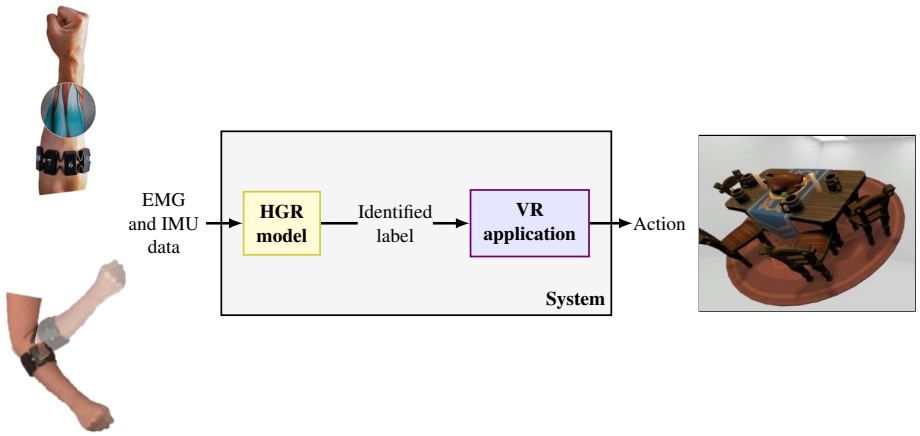


Fig. 2. System.

Figure 2 shows the general purpose of this work, where the HGR model and the application complement each other as a system that combines the potential of the HGR field applied to a VR application. The rest of the paper is organized as follows. Section II describes the works related to gesture recognition and their applications. The dataset, the structure of the HGR model, the description of the application, and integration of both components are explained in Section III. Experimental results and their analysis are presented in Section IV. Finally, Section V concludes this work.

2 Related Works

The development of HGR models is used to perform different tasks including but not limited to motor control, prosthetic device control, and hand motion classification [33]. Gesture recognition applied to VR has a wide array of fields, such as vehicle driving simulation [29, 32], games [28, 20], navigation of maps [16], sign language gestures [26], among others.

The most frequent machine learning algorithms applied to gesture recognition based on EMG and IMU data are artificial neural network-based algorithms [11, 22, 8, 34], classifier-based algorithms [18, 25, 17], and linear discriminant analysis [13, 31]. According to the literature review, for this work we choose the kNN classification algorithm because it is one of the simplest and most optimal models in terms of classification.

The use of Myo Armband and a virtual environment was already evaluated in [10] through SUS, which gives us an idea of the usability qualification, however the overall result of [10] is lower than the obtained in this article. In addition, according to [21], it is easier to use the Myo Armband in the control of tasks related with daily life. Likewise, Rawat[21] affirms that users can interact with various applications simply by making our hands work, which will be demonstrated in this article.

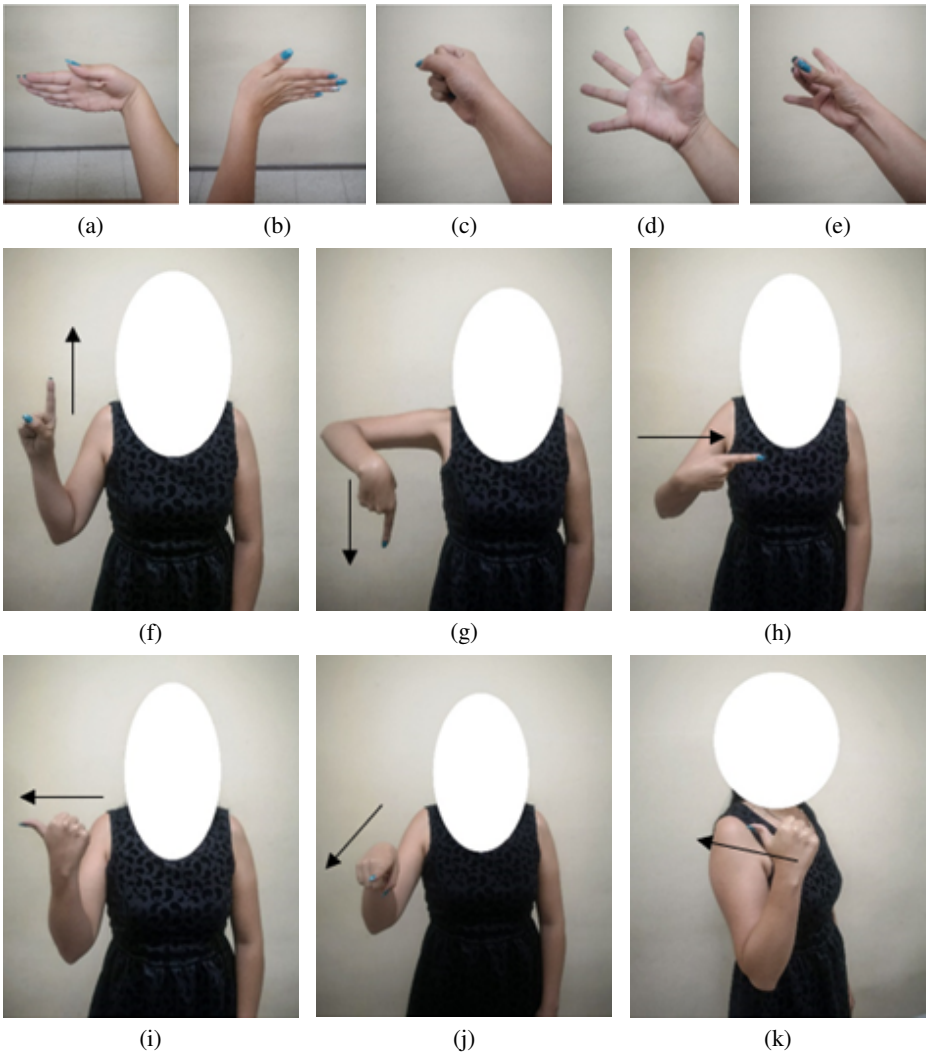


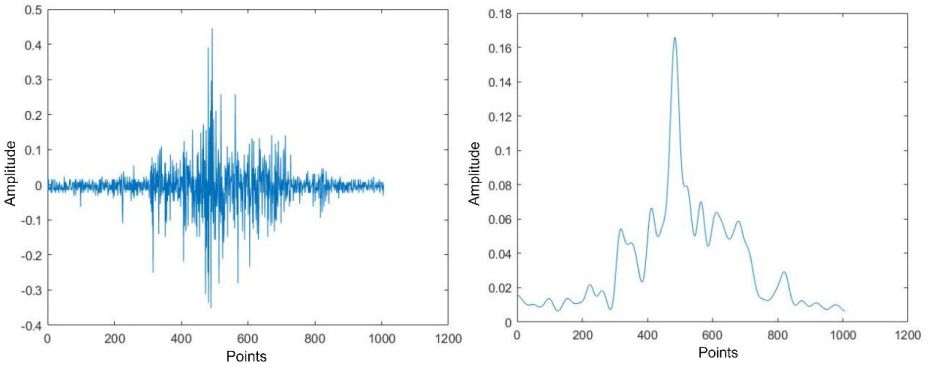
Fig. 3. Gestures used to the Dataset (a) waveIn, (b) waveOut, (c) fist, (d) open, (e) pinch, (f) up, (g) down, (h) left, (i) right, (j) forward and (k) backward.

3 Methodology

In this section, we describe the dataset and structure of HGR model and the VR application.

3.1 Dataset and HGR Model

This subsection describes the dataset and the HGR model proposed in this work. The HGR model is composed by the following steps: data acquisition, preprocessing, feature extraction and classification.



(a) (b)

Fig. 4. (a) raw EMG signal (b) preprocessed EMG signal.

Dataset. EMG and IMU signals from 85 users are included in the dataset, which can be found in [2]. The signals were gathered while performing a set of 12 gestures, which includes the 11 gestures specified in Figure 3 (wave in, wave out, fist, open, pinch, up, down, left, right, forward, and backward), along with an additional gesture denoted as the "relax" gesture. Each user performed a total of 180 samples by executing 15 repetitions of 5-second intervals for each gesture.

Data acquisition. The sliding window technique was applied to acquire the input data for the classifier. We used a window of $N = 480$ points for the classification of EMG signals and another window of $M = 480$ points for IMU signals. The complete signal was analyzed, in both cases, with the same number of iterations using the stride of $L = 200$.

Preprocessing. The purpose of this stage is to facilitate the subsequent phases of feature extraction and classification. It was applied only to EMG signals because they have an irregular appearance, as shown in Figure 4a. Preprocessing consists of rectifying and filtering the window N. For rectification, the absolute value of both windows was calculated. Then, for filtering, a fourth-order Butterworth filter with and a cutoff frequency of 5 Hz to reduce the noise was applied. Figure 4b shows the result of the preprocessing as a smoother EMG signal.

Feature extraction. We work with EMG windows to classify static gestures since the execution of these gestures is dominated by muscle activity, which is measured using EMG signals. Similarly, we use IMU windows to classify dynamic gestures which are dominated by arm movement and measured by IMU signals.

In this stage, the two EMG and IMU sliding windows are analyzed in order to select one window for the classification phase. For this purpose, the "energy" feature extraction function is applied over both type sliding windows. It measures the energy distribution of both signals [6]. The output of this function is a feature vector used as input to a switch, as shown in Figure 5. The switch is based on a logistic linear classification model.

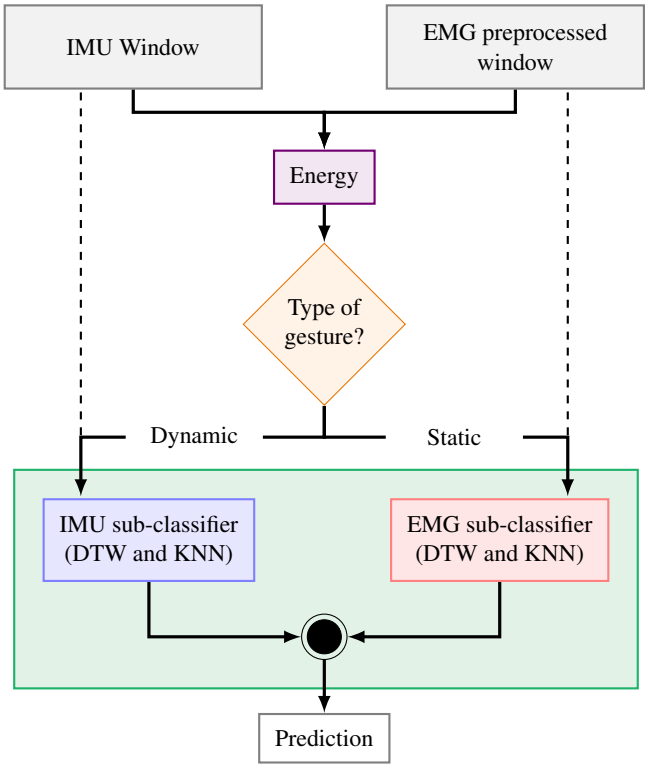


Fig. 5. Feature extraction and classification stages.

This model determines whether a pair of EMG and IMU windows of the same signal corresponds to a static or dynamic gesture. The model was trained using the same "energy" function for the training dataset. Its evaluation achieved an average classification accuracy rate of 93.02%. Through this switch, a feature window is chosen according to the type of gesture determined and, consequently, the subclassifier corresponding to the feature window (EMG or IMU).

Classification. The classification module works exclusively with EMG signals or with IMU signals separately. Consequently, the classifier is constituted by 2 sub-classifiers as shown in Figure 5: the EMG sub-classifier which classifies static gestures and the IMU sub-classifier which classifies dynamic gestures. Both subclassifiers were built using the KNN algorithm.

The approach on which KNN is based in this work is the estimation of the conditional probability based on the relative frequency of the nearest neighbors to the window to be classified. For both subclassifiers, the value of k nearest neighbors was determined with the formula $k = \text{ceil}[\log_2 N]$, where N is the number of samples composing the dataset corresponding to the subclassifier. The threshold was set at 80%, to avoid false positives. The code of the proposed HGR model was written with MATLAB version 2021b and is publicy available in [3].

Table 1. Technical specifications of the application development environment.

Aspect	Specification
Unity Version	2021.2.3f1
GPU	NVIDIA GeForce GTX 1650
CPU	Intel Core i7-10700F
RAM	16 GB
HDD	932 GB
SSD	233 GB
Motherboard	Prime H410M-E

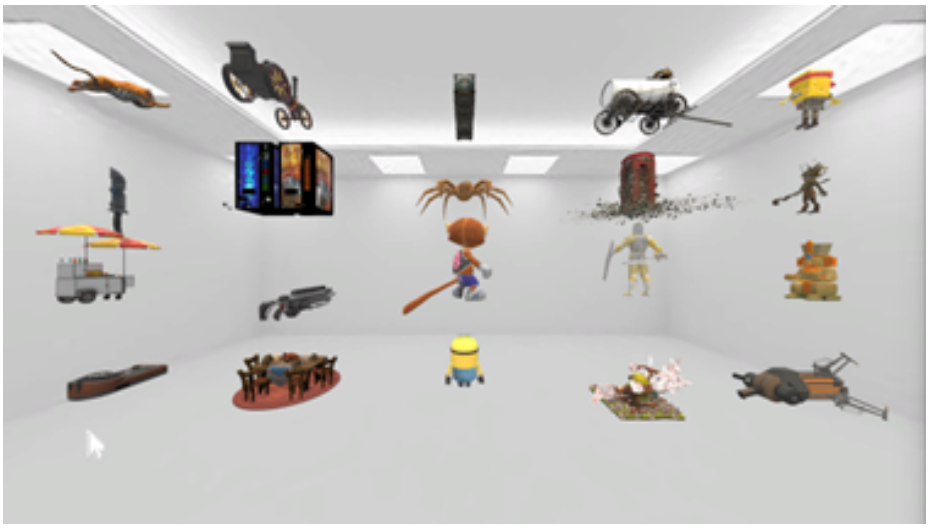


Fig. 6. 3D model viewer "Gallery" scene.

3.2 3D Model Viewer

To demonstrate the feasibility of the model, a software application (3D Model Viewer) controlled by the HGR system is developed. The 11 hand gestures are used to execute interaction actions with models within the application, such as: moving between models, moving the model, accepting an action, among others. The gestures are detected thanks to the use of a recognition model detailed above.

The methodology for the 3D model viewer uses the waterfall model. [24] . The phases of the methodology include: requirements specification, design, development, verification and validation, and operation of the application. The requirements specification phase collected user needs through meetings with the end user. This phase expresses the user's intention in 13 use cases, as shown in Annex B.

In the design phase, the space names, scenes, class diagram, and complements were described. This phase expresses the architecture of the application through the class diagram, as observed in Annex A. Likewise, in the development phase, the source code was implemented using C# as the programming language.

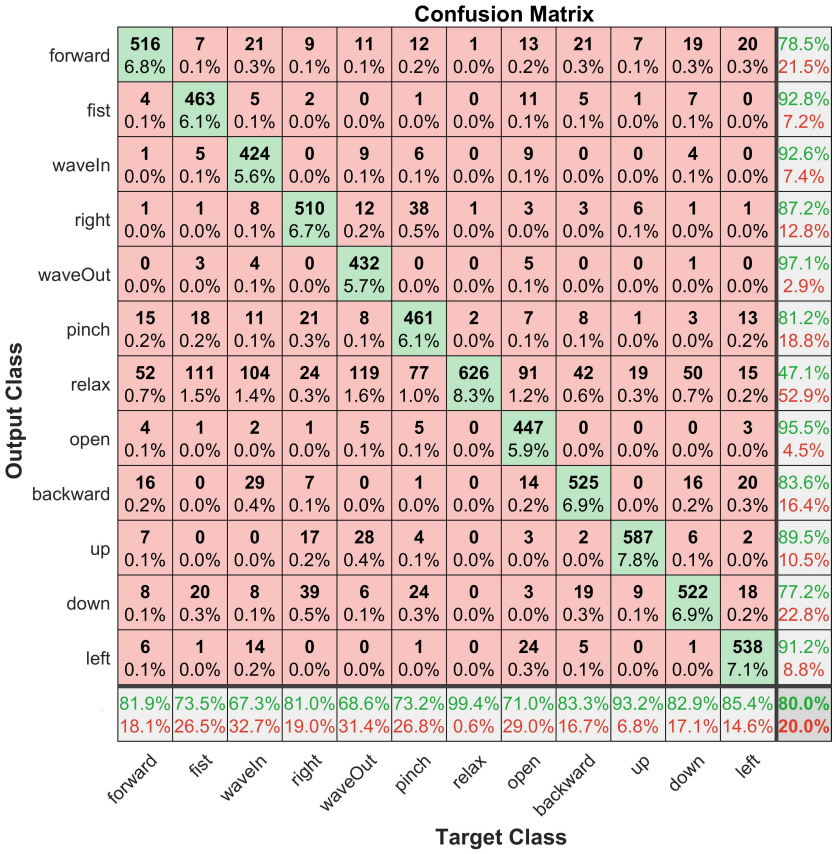


Fig. 7. Confusion matrix for proposed model.

Among other tools for coding the viewer, the IDE VS 2022 version 17.2 and Unity version 2021.2.3f1 were used. Additional technical specifications for the development are described in the Table 1. To comply with the verification and validation phase, integration and acceptance tests were carried out. The integration tests were ascending to perform evaluations of the modules from lower to higher levels. For its part, the acceptance tests are based on each use case specified in Annex B. Finally, in the operation phase, the application was compiled with support for virtual reality in the 64-bit Windows 10 operating system. One of the scenes (Gallery) of the operational application is shown in Figure 6.

3.3 Integration

Once the developed application operational, communication between the application development environment and the recognition model was established. This integration used socket technology where Unity and Matlab took the roles of server and client, respectively.

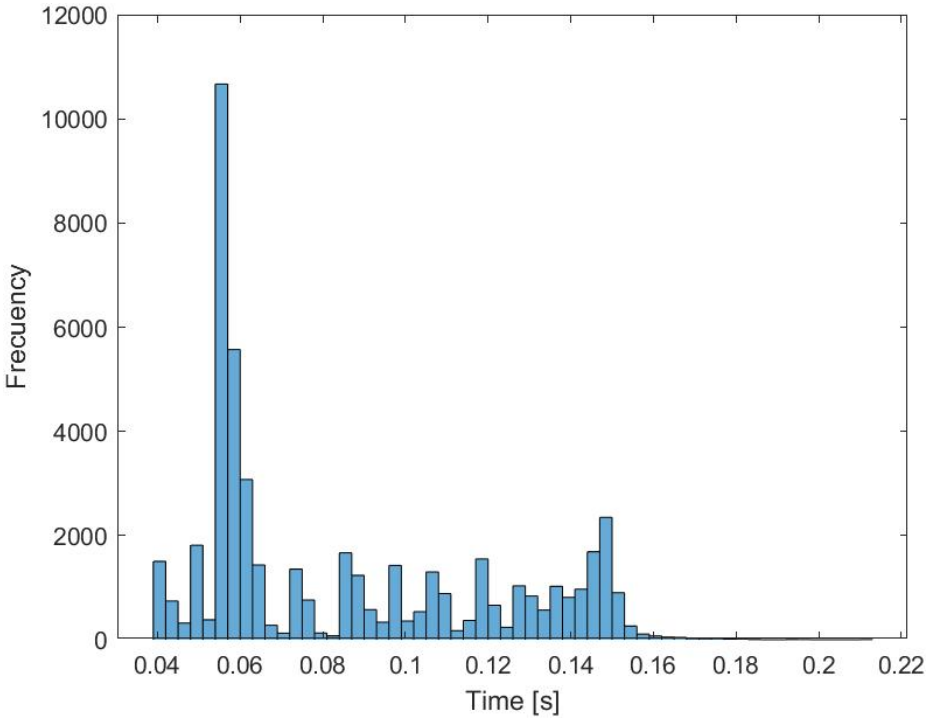


Fig. 8. Histogram of the processing time of each window observation.

The IP address used was "127.0.0.1", the port was 55001 and 10 seconds as time out. The source code of the proposed HGR model and the VR application is publicy available in [3]. A video demonstration of the execution of the complete system can be found in [4].

4 Results

In this section, we present the results of the tests applied to the HGR model and the system.

4.1 Performance of the HGR Model

We evaluate the performance of HGR model with 42 users of dataset described in Section III. Classification results are shown in the Figure 7. The label that presents the highest sensitivity is Up (93.2%) while the label with the lowest sensitivity is WaveIn (67.3%).

On the other hand, the labels with the highest and lowest precision are WaveOut (97.1%) and Down (77.2%), respectively. It is also illustrated that the overall classification accuracy of the model is 80% while the average accuracy is 84.45%.

Table 2. Result of SUS.

Person\Item	1	2	3	4	5	6	7	8	9	10	**Total**
1	5	0	0	0	4	0	5	0	5	0	**97,5**
2	5	1	5	2	5	0	5	0	5	2	**100**
3	4	0	4	2	5	0	4	0	4	1	**95**
4	5	0	4	4	5	0	5	0	5	0	**100**
5	5	0	4	0	5	5	5	1	5	0	**95**
6	3	1	4	1	3	3	4	1	5	4	**72,5**
7	3	3	4	2	1	1	3	4	3	1	**57,5**
8	3	3	3	0	3	1	3	3	5	0	**75**
9	1	0	4	1	2	1	4	1	5	0	**82,5**
10	4	2	4	4	4	2	2	2	3	5	**55**

It is important to note that the results obtained in terms of classification (80% accuracy) taking into account the high number of labels to classify (11 gestures shown in Figure 3) are considered highly satisfactory for our RV application. The response time is defined as the time that each sub-classifier (EMG or IMU) takes to process and classify each window of the signal.

The processing time of each window was measured and stored for later analysis. Figure 8 shows the processing times for each processed window. The highest processing time is 0.22 seconds while most of measures remains below 0.16 seconds. In conclusion, our HGR model returns a real time response below 0.3 seconds, which it can be considered as real time.

4.2 Usability Test Results

The system was subjected to the System Usability Scale (SUS). The evaluation was carried out by 10 participants between 21 and 37 years old. The 10 statements received a rating between 0 and 5. The rating of 0 is for the evaluator to show that they totally disagree while rating 5 is to show total agreement. The results of the 10 users for each statement are shown in Table 2. To calculate the global rating of the application for each user, the following formula was used:

$$t = (x - 5) + 2.5(25 - y), \tag{1}$$

where:

$x =$ Sum of even item responses.
$y =$ Sum of odd item responses.
$t =$ Global.

The average of the global was 83. According to [15], the usability score of the system is Excellent. The obtained score shares the acceptable range as well as many other applications described in [15], such as Excel, Gmail, among others.

5 Conclusions and Future Work

A HGR model based on EMG and IMU signals together with a VR application has been presented. The structure of the developed HGR model is based on the stages of data acquisition, preprocessing, feature extraction and classification.

The EMG-IMU-EPN-100+ was used to evaluate the model, and the results show a classification accuracy of $80.04\% \pm 13.66\%$ and a recognition accuracy of $66.12\% \pm 18.30\%$. The response time of the model is below 0.22 seconds which, according to the literature, validates the real-time performance of the model. Similarly, the VR application was successfully integrated with the HGR model. A global average of 83 in SUS scale demonstrate the acceptable range of the system.

Future work on the HGR model includes testing with a stride between consecutive windows of less than 1 second. Similarly, the use of parallel processing for the classification step is also a potential improvement. For its part, the 3D Model Viewer could implement wireless virtual reality glasses, improving the user's mobility. The application could also provide model management tasks from its graphical interface.

Acknowledgments. The authors gratefully acknowledge the financial support provided by the Escuela Politécnica Nacional (EPN) for the development of the research project "PIGR-22-09 Avances para el desarrollo de un prototipo de prótesis mioeléctrica de mano y control avanzado de su operación usando Inteligencia Artificial".

References

1. Ahmed, S., Kallu, K. D., Ahmed, S., Cho, S. H.: Hand gestures recognition using radar sensors for human-computer-interaction: A review. Remote Sensing, vol. 13, no. 3, pp. 527 (2021) doi: 10.3390/rs13030527
2. Artificial Intelligence and Computer Vision Research Lab Alan Turing: Dataset EMG-IMU-EPN-100+ (2022)
3. Artificial Intelligence and Computer Vision Research Lab Alan Turing: Source code of the design and application of a recognition model of 11 hand gestures using EMG, IMU, DTW and KNN (2022)
4. Artificial Intelligence and Computer Vision Research Lab Alan Turing: Video demo of the design and application of a recognition model of 11 hand gestures using EMG, IMU, DTW and KNN (2022)
5. Arun-Faisal, I., Waluyo-Purboyo, T., Raharjo-Ansori, A. S.: A review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture. Journal of Engineering and Applied Sciences, vol. 15, no. 3, pp. 826–829 (2019) doi: 10.36478/jeasci.2020.826.829
6. Barona-López, L. I., Valdivieso-Caraguay, A. L., Vimos, V. H., Zea, J. A., Vásconez, J. P., Álvarez, M., Benalcázar, M. E.: An energy-based method for orientation correction of EMG bracelet sensors in hand gesture recognition systems. Sensors, vol. 20, no. 21, pp. 6327 (2020) doi: 10.3390/s20216327
7. Benalcázar, M. E., Jaramillo, A. G., Jonathan, Zea, A., Páez, A., Andaluz, V. H.: Hand gesture recognition using machine learning and the Myo armband. In: 25th European Signal Processing Conference (EUSIPCO), pp. 1040–1044 (2017) doi: 10.23919/EUSIPCO.2017.8081366

8. Can, C., Kaya, Y., Kılıç, F.: A deep convolutional neural network model for hand gesture recognition in 2d near-infrared images. Biomedical Physics and Engineering Express, vol. 7, no. 5, pp. 55005 (2021) doi: 10.1088/2057-1976/ac0d91
9. Chirinos-Delfino, Y.: La realidad virtual como mediadora de aprendizajes: Desarrollo de una aplicación móvil de realidad virtual orientada a niños. Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, vol. 1, no. 30, pp. 136–137 (2021) doi: 10.24215 /18509959.30.e16
10. De-Paolis, L. T., De-Luca, V.: The impact of the input interface in a virtual environment: The Vive controller and the Myo armband. Virtual Reality, vol. 24, no. 3, pp. 483–502 (2020) doi: 10.1007/s10055-019-00409-6
11. Djemal, A., Hellara, H., Barioul, R., Atitallah, B. B., Ramalingame, R., Fricke, E., Kanoun, O.: Real-time model for dynamic hand gestures classification based on inertial sensor. In: IEEE 9th International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications, pp. 1–6 (2022) doi: 10.1109/CIVEMSA53371 .2022.9853648
12. Jaramillo-Yánez, A., Benalcázar, M. E., Mena-Maldonado, E.: Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review. Sensors, vol. 20, no. 9, pp. 2467 (2020) doi: 10.3390/s20092467
13. Khushaba, R. N., Nazarpour, K.: Decoding HD-EMG signals for myoelectric control - how small can the analysis window size be? IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 8569–8574 (2021) doi: 10.1109/LRA.2021.3111850
14. Koch, P., Dreier, M., Maass, M., Böhme, M., Phan, H., Mertins, A.: A recurrent neural network for hand gesture recognition based on accelerometer data. In: 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 5088–5091 (2019) doi: 10.1109/EMBC.2019.8856844
15. Kortum, P. T., Bangor, A.: Usability ratings for everyday products measured with the system usability scale. International Journal of Human-Computer Interaction, vol. 29, no. 2, pp. 67–76 (2013) doi: 10.1080/10447318.2012.681221
16. Lee, Y. S., Sohn, B. S.: Immersive gesture interfaces for navigation of 3D maps in HMD-based mobile virtual environments. Mobile Information Systems, vol. 2018 (2018) doi: 10.1155/2018/2585797
17. Lian, K. Y., Chiu, C. C., Hong, Y. J., Sung, W. T.: Wearable armband for real time hand gesture recognition. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 2992–2995 (2017) doi: 10.1109/SMC.2017.8123083
18. Maragliulo, S., Lopes, P. F. A., Osório, L. B., De Almeida, A. T., Tavakoli, M.: Foot gesture recognition through dual channel wearable EMG system. IEEE Sensors Journal, vol. 19, no. 22, pp. 10187–10197 (2019) doi: 10.1109/JSEN.2019.2931715
19. Nandwana, B., Tazi, S., Trivedi, S., Kumar, D., Vipparthi, S. K.: A survey paper on hand gesture recognition. In: 7th International Conference on Communication Systems and Network Technologies, pp. 147–152 (2017) doi: 10.1109/CSNT.2017.8418527
20. Rautaray, S. S., Agrawal, A.: Interaction with virtual game through hand gesture recognition. In: International Conference on Multimedia, Signal Processing and Communication Technologies, pp. 244–247 (2011) doi: 10.1109/MSPCT.2011.6150485
21. Rawat, S., Vats, S., Kumar, P.: Evaluating and exploring the MYO ARMBAND. In: International Conference System Modeling and Advancement in Research Trends, pp. 115–120 (2016) doi: 10.1109/SYSMART.2016.7894501
22. Riaz, M. M., Zhang, Z.: Surface EMG real-time chinese language recognition using artificial neural networks. In: Intelligent Life System Modelling, Image Processing and Analysis, Springer Singapore, pp. 114–122 (2021) doi: 10.1007/978-981-16-7207-1_12

23. Slater, M., Wilbur, S.: A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. Presence: Teleoperators and Virtual Environments, vol. 6, no. 6, pp. 603–616 (1997) doi: 10.1162/pres.1997.6.6.603
24. Sommerville, I.: Software Engineering. Pearson (2011)
25. Tavakoli, M., Benussi, C., Lopes, P. A., Osorio, L. B., de Almeida, A. T.: Robust hand gesture recognition with a double channel surface EMG wearable armband and SVM classifier. Biomedical Signal Processing and Control, vol. 46, pp. 121–130 (2018) doi: 10.1016/j.bspc.2018.07.010
26. Vaitkevičius, A., Taroza, M., Blažauskas, T., Damaševičius, R., Maskeliūnas, R., Woźniak, M.: Recognition of american sign language gestures in a virtual reality using leap motion. Applied Sciences, vol. 9, no. 3, pp. 445 (2019) doi: 10.3390/app9030445
27. Vatavu, R. D., Pentiuc, S. G.: Multi-level representation of gesture as command for human computer interaction. Computing and Informatics, vol. 27, no. 6, pp. 837–851 (2008)
28. Wen, F., Sun, Z., He, T., Shi, Q., Zhu, M., Zhang, Z., Li, L., Zhang, T., Lee, C.: Machine learning glove using self-powered conductive superhydrophobic triboelectric textile for gesture recognition in VR/AR applications. Advanced Science, vol. 7, no. 14 (2020) doi: 10.1002/advs.202000261
29. Xu, D.: A neural network approach for hand gesture recognition in virtual reality driving training system of SPG. In: 18th International Conference on Pattern Recognition, vol. 3, pp. 519–522 (2006) doi: 10.1109/ICPR.2006.109
30. Yeo, H. S., Lee, B. G., Lim, H.: Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. Multimedia Tools and Applications, vol. 74, no. 8, pp. 2687–2715 (2015) doi: 10.1007/s11042-013-1501-1
31. Young, A. J., Smith, L. H., Rouse, E. J., Hargrove, L. J.: Classification of simultaneous movements using surface EMG pattern recognition. IEEE Transactions on Bio-Medical Engineering, vol. 60, no. 5, pp. 1250–1258 (2013) doi: 10.1109/tbme.2012.2232293
32. Young, G., Milne, H., Griffiths, D., Padfield, E., Blenkinsopp, R., Georgiou, O.: Designing mid-air haptic gesture controlled user interfaces for cars. In: Proceedings of the ACM on Human-Computer Interaction, vol. 4, pp. 1–23 (2020) doi: 10.1145/3397869
33. Zaman-Khan, R., Ibraheem, N.: Hand gesture recognition: A literature review. International Journal of Artificial Intelligence and Applications, vol. 3, no. 4, pp. 161–174 (2012) doi: 10.5121/ijaia.2012.3412
34. Zhou, C., Yang, L., Liao, H., Liang, B., Ye, X.: Ankle foot motion recognition based on wireless wearable sEMG and acceleration sensors for smart AFO. Sensors and Actuators A: Physical, vol. 331, pp. 113025 (2021) doi: 10.1016/j.sna.2021.113025

ISSN 1870-4069

Penalty Functions to Improve the Performance of MOEA's for Portfolio Optimization Problems

Lourdes Uribe, Uriel Trejo-Ramirez,
Yael Andrade-Ibarra, Oliver Cuate,
Victor Cordero

Instituto Politecnico Nacional,
Escuela Superior de Física y Matemáticas,
Mexico

{luriber, ocuateg}@ipn.mx, {utrejor1700, yandradei1600,
vcorderoc1600}@alumno.ipn.mx

Abstract. With the significant growth of the financial market, investment options have increased, which can pose a challenge. Thus, one of the most studied problems in the financial field is the Portfolio Optimization Problem, where one seeks the major possible return but with minimal risk. Given that both objectives are in conflict and must be simultaneously optimized, a multi-objective optimization problem (MOP) arises naturally. Even more, since certain conditions must be satisfied, this MOP is restricted; thus, we really are dealing with a constrained MOP (CMOP). Multi-objective evolutionary algorithms (MOEAs) are a widely accepted approach for the numerical treatment of these problems. For constrained problems, however, these methods still have room for improvement to compute satisfactory approximations of the solution sets. In this work, we propose to use different penalty strategies to improve NSGA-II and NSGA-III performance when dealing with the portfolio optimization problem. We claim that penalty strategies helped the evolutionary algorithm to obtain a greater number of feasible individuals while preserving optimal solutions. Numerical results support this claim.

Keywords: Portfolio optimization, penalization, evolutionary algorithms.

1 Introduction

With the significant growth of the financial market, investment options have increased, which can pose a challenge. The availability of numerous options in the market makes it difficult to decide which is the best, even if there is always a single best option. We must remember that every investment comes with risk. If we analyze it carefully, we can identify two different objectives when investing: on the one hand, we aim to maximize investment returns, and on the other hand, we seek to minimize the associated risk.

These objectives often conflict since higher expected returns typically come with higher risks. Such problems are known as multi-objective optimization problems (MOPs). Several approaches have been explored to address these types of problems, commonly involving the application of computational tools.

Algorithm 1 Quadratic Penalty Method.

Require: Given $\mu_0 > 0$, a nonnegative sequence $\{\tau_k\}$ with $\tau_k \to 0$, and a starting point $\boldsymbol{x}_0^s$;
  for $k = 0, 1, 2, \ldots$ **do**
    Find an approximate minimizer $\boldsymbol{x}_k$ of Q as in Equation 5, starting at $\boldsymbol{x}_k^s$, and finishing when $\|\nabla Q(\boldsymbol{x})\| \leq \tau_k$;
    if convergence test is satisfied **then**
      stop **return** approximate solution $\boldsymbol{x}_k^s$
    end if
    Choose new penalty parameter $\mu_{k+1} > \mu_k$;
    Choose new starting point $\boldsymbol{x}_{k+1}^s$;
  end for

One of these tools is multi-objective algorithms, also known as MOEAs (Multi-Objective Evolutionary Algorithms), which employ techniques inspired by biological evolution to find optimal solutions. MOEAs have caught the interest of many researchers (see, e.g., [8, 5, 3, 9, 12]) over the last decades. Some reasons for this include that MOEAs are of global nature.

Moreover, due to their global approach, they compute a finite size approximation of the entire Pareto Set in one single execution of the algorithm. Also, they have been successfully applied in several applications [18, 20, 29, 24], particularly in the portfolio optimization problem [30, 14].

However, not all of these algorithms handle constraints efficiently. Most MOEAs use feasibility rules to deal with constrained MOPs [7, 15, 21], while others use penalty strategy [25]. Penalty strategy involves assigning a penalty value to infeasible solutions based on the degree of violation. Therefore in the search for optimal solutions, these infeasible solutions will be left behind since they will not get the minimal objective value due to the imposed penalization.

Various families of penalty functions have been studied to improve MOEAs performance when dealing with constrained optimization. There are two main approaches: the first one is based on the constraint violation value, and the second one is based on the distance to the feasible region [13, 25]. One of the most common problems when using both approaches is that the search's effectiveness strongly depends on the selected penalty function.

While evolutionary-guided search with adaptive penalization demonstrates an advantage as an optimization method for these highly restrictive problems [6, 19]. Utilizing feedback from solution search, as well as any specific information, provides an adaptive and dynamic penalization that is effective. The portfolio optimization problem aims to find the optimal distribution of financial assets to maximize expected return and minimize risk. MOEAs provide an effective solution to this problem due to their ability to work with multiple objectives and find optimal points.

The traditional portfolio optimization approach is based on Markowitz's theory. However, this approach assumes normal distributions for asset returns, which can be an idealistic scenario. Additionally, Markowitz's theory does not consider the diversity of objectives that investors may have. Utilization of MOEAs becomes crucial in this context. For example, particle swarm optimization (PSO) has been successfully used in different portfolio optimization problems [30, 14].

Algorithm 2 Classical l_1 Penalty method.

Require: Given $\mu_0 > 0$, tolerance $\tau > 0$ and a starting point $\boldsymbol{x}_0^s$;
  for $k = 0, 1, 2, \ldots$ **do**
    Find an approximate minimizer $\boldsymbol{x}_k^s$ of $\phi_1(x)$, starting at $\boldsymbol{x}_k^s$;
    if $\mathrm{MInf}(\boldsymbol{x}) < \tau$ **then**
      Stop **return** approximate solution $\boldsymbol{x}_k^s$
    end if
    Choose new penalty parameter $\mu_{k+1} > \mu_k$;
    Choose new starting point $\boldsymbol{x}_{k+1}^s$;
  end for

Also, ant colony optimization has been applied to Markowitz's portfolio model [11]. In [2], the authors applied the fireworks algorithm to solve the constrained portfolio problem for the first time. Also, genetic algorithms have been used to solve this problem, specifically in [1] NSGA-II and NSGA-III were employed to solve the portfolio problem for 2 and 3 objectives.

Here, the authors presented that NSGA-II was effective only for two objectives and that NSGA-III was effective only for three objectives. This work aims to optimize various investment portfolios using three different penalty methods implemented on NSGA-II and NSGA-III algorithms.

A comparative analysis is conducted between the results obtained by the MOEA without a penalty and those obtained using the different penalty strategies. Based on the results, we show that when dealing with the portfolio optimization problem, it is very important to implement the correct penalty strategy, as it improves the normal behavior of MOEAs in this problem, especially NSGA-II and NSGA-III.

2 Background

Here, we consider continuous MOPs that can be expressed as:

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathbb{R}^n} \boldsymbol{F}(\boldsymbol{x}), \\ \text{s.t. } \; &g_i(\boldsymbol{x}) \leq 0 \quad \text{for} \quad i = 1, \ldots, m, \\ &h_i(\boldsymbol{x}) = 0 \quad \text{for} \quad i = 1, \ldots, q. \end{aligned} \tag{1}$$

Hereby, $\boldsymbol{F}$ is the map of objective functions $\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x}))^T$. Each objective $f_i : \mathbb{R}^n \to \mathbb{R}$ is assumed for simplicity to be continuously differentiable, and with feasible domain:

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n : h_i(\boldsymbol{x}) = 0, i = 1, \ldots, q \text{ and } g_i(\boldsymbol{x}) \leq 0, \; i = 1, \ldots, m\}. \tag{2}$$

The optimality of a MOP is defined using the concept of Pareto dominance: let $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^k$, then $\boldsymbol{v}$ is less or equal than $\boldsymbol{w}$ ($\boldsymbol{v} \leq_p \boldsymbol{w}$), if $v_i \leq w_i$ for all $i \in \{1, \ldots, k\}$; the relation $<_p$ is defined analogously. A vector $\boldsymbol{y} \in \Omega$ is dominated by a vector $\boldsymbol{x} \in \Omega$ ($\boldsymbol{x} \prec \boldsymbol{y}$) with respect to (1) if $\boldsymbol{F}(\boldsymbol{x}) \leq_p \boldsymbol{F}(\boldsymbol{y})$ and $\boldsymbol{F}(\boldsymbol{x}) \neq \boldsymbol{F}(\boldsymbol{y})$, else $\boldsymbol{y}$ is called non-dominated by $\boldsymbol{x}$.

Algorithm 3 Augmented Lagrangian Method.

Require: Given $\mu_0 > 0$, tolerance $\tau > 0$, starting points $\boldsymbol{x}_0^s$ and λ^0;
for $k = 0, 1, 2, \ldots$ **do**
Find an approximate minimizer $\boldsymbol{x}_k^s$ of $\mathcal{L}_A(\cdot, \lambda^k)$, starting at $\boldsymbol{x}_k^s$, and finishing when $\|\nabla \mathcal{L}_A(\boldsymbol{x}_k; \lambda^k)\| \leq \tau_k$;
if convergence test is satisfied **then**
Stop **return** approximate solution $\boldsymbol{x}_k^s$
end if
Update Lagrange multipliers using equation 9 to obtain λ^{k+1};
Choose new penalty parameter $\mu_{k+1} \geq \mu_k$;
Set starting point for the next iteration to $\boldsymbol{x}_{k+1}^s = \boldsymbol{x}_k$;
Select tolerance τ_{k+1};
end for

In case $\boldsymbol{F}(\boldsymbol{x}) <_p \boldsymbol{F}(\boldsymbol{y})$ the relation is called strong Pareto dominance. A point $\boldsymbol{x}^* \in \mathbb{R}^n$ is Pareto optimal to (1) if there is no $\boldsymbol{y} \in \Omega$ which dominates $\boldsymbol{x}$. The set of all the Pareto optimal points P_Ω is called the Pareto set, and its image $\boldsymbol{F}(P_\Omega)$ is called the efficient set or Pareto front.

2.1 Portfolio Optimization Problem

The portfolio model, also known as the Markowitz model, aims to maximize the return function while minimizing the risk function. Therefore, a MOP naturally arises. We can define the problem as:

$$\begin{aligned} \text{Max. Return:} \quad & \sum_{i=1}^{N} \boldsymbol{w}_i \mu_i, \\ \text{Min. Risk:} \quad & \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{w}_i \boldsymbol{w}_j \sigma_{ij}, \\ \text{s.t.} \quad & \sum_{i=1}^{N} \boldsymbol{w}_i = 1, \\ & 0 \leq \boldsymbol{w}_i \leq 1 \quad \text{for} \quad i = 1, \ldots, N, \end{aligned} \tag{3}$$

where N is the number of available assets, μ_i represents the expected return of asset i, σ_{ij} is the covariance between assets i and j, and $\boldsymbol{w}_i$ is the decision variable for asset i. It is worth noticing that w_i has a weighting effect on the return function and the covariance matrix; for more details see [27].

As mentioned before, we try to find solutions that simultaneously satisfy the above conflicting functions. In this work, the optimal portfolio will be the one that provides us with maximum return and minimum risk.

Table 1. This table presents the MOEAs parameters used in the experimental section.

Parameter	NSGAII	NSGAIII
Population size	100	100
Crossover probability	0.9	1
Mutation probability	0.1	1/n
Distribution index for crossover	20	20
Distribution index for mutation	30	20

When diversification is considered, the model can be written as:

$$\begin{aligned} \text{Max.} \quad \text{Return:} \quad & \sum_{i=1}^{N} \boldsymbol{w}_i \mu_i - \sum_{i=1}^{N} c_i \mid \boldsymbol{w}_i - \boldsymbol{w}_i^0 \mid, \\ \text{Min.} \quad \text{Risk:} \quad & \sum_{i=1}^{N} \sum_{j=1}^{N} \boldsymbol{w}_i \boldsymbol{w}_i \sigma_{ij}, \\ \text{Max.} \quad \text{Entropy:} \quad & \sum_{i=1}^{N} \boldsymbol{w}_i \log(\boldsymbol{w_i}), \\ \text{s.t.} \quad & \sum_{i=1}^{N} \boldsymbol{w}_i = 1, \\ & 0 \leq \boldsymbol{w}_i \leq 1 \quad \text{for} \quad i = 1, \ldots, N, \end{aligned} \tag{4}$$

where $\boldsymbol{w}^0$ is the existing portfolio and $\sum_{i=1}^{N} c_i \mid \boldsymbol{w}_i - \boldsymbol{w}_i^0 \mid$ is the total transaction cost of the portfolio. Here, entropy is used as the divergence measure of asset portfolio in finance literature [17].

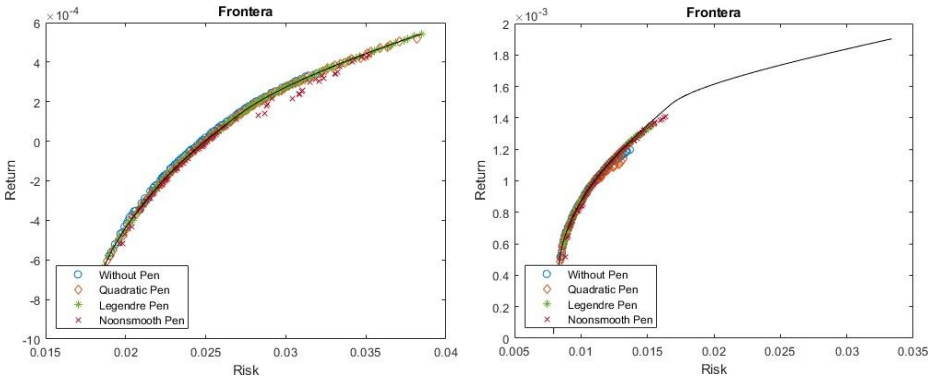
2.2 Penalty Methods

– **Quadratic Penalty Method.** In this method, the penalty terms are the squares of the constraint violations. We define the quadratic penalty function for Problem 1 as:

$$Q(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{\mu}{2} \sum_{i=1}^{q} h_i^2(\boldsymbol{x}) + \frac{\mu}{2} \sum_{i=1}^{m} (\max\{g_i(\boldsymbol{x}), 0\})^2, \tag{5}$$

where $\mu > 0$ is the penalty parameter. In Algorithm 1, the general framework based on the quadratic penalty function is presented. It is worth noticing that the parameter sequence $\{\mu_k\}$ can be chosen adaptively, considering the difficulty of minimizing the penalty function at each iteration.

– **Nonsmooth Penalty Function.** Nonsmooth penalty functions are less dependent on the strategy used to choose penalty parameters, which makes them desirable.



(a) (b)

Fig. 1. Comparison of the obtained Pareto fronts for Portfolio 1 and Portfolio 3, respectively, on a certain execution.

A popular nonsmooth penalty function for the general nonlinear programming problem is the l_1 penalty function, which can be defined as:

$$\phi_1(\boldsymbol{x}) = f(\boldsymbol{x}) + \mu \sum_{i=1}^{q} |h_i(\boldsymbol{x})| + \mu \sum_{i=1}^{m} \max\{g_i(\boldsymbol{x}), 0\}, \tag{6}$$

where $\mu > 0$ is the penalty parameter. Note that $\phi_1(x)$ is not differentiable at some $\boldsymbol{x}$ because of the absolute value and $\| \cdot \|$ function. Despite not being differentiable, Equation 6 has a directional derivative along any direction, which allows defining a stationary point of the measure of infeasibility as:

$$\mathrm{MInf}(\boldsymbol{x}) = \sum_{i=1}^{q} |h_i(\boldsymbol{x})| + \sum_{i=1}^{m} \max\{g_i(\boldsymbol{x}), 0\}, \tag{7}$$

When $\mathrm{MInf}(\boldsymbol{x})$ tends to zero, it indicates feasibility. Algorithm 2 presents a general framework based on the l_1 penalty function. Exact nonsmooth penalty functions can be defined in terms of other norms, see [23].

– **Augmented Lagrangian Method: Equality Constraints.** This algorithm is similar to the quadratic penalty algorithm, but it reduces the likelihood of ill-conditioning by introducing Lagrange multipliers into the function. This function is known as the augmented Lagrange function, which preserves smoothness; unlike Nonsmooth penalty functions, the augmented Lagrange function largely preserves smoothness. By definition:

$$\mathcal{L}_A(\boldsymbol{x}, \lambda) = f(x) - \sum_{i=1}^{q} \lambda_i h_i(\boldsymbol{x}) + \frac{\mu}{2} \sum_{i=1}^{q} h_i^2(\boldsymbol{x}), \tag{8}$$

where:

$$\lambda_i^{k+1} = \lambda_i^k - \mu_k h_i(\boldsymbol{x}_k), \ \forall\, i = 1, \ldots, q. \tag{9}$$

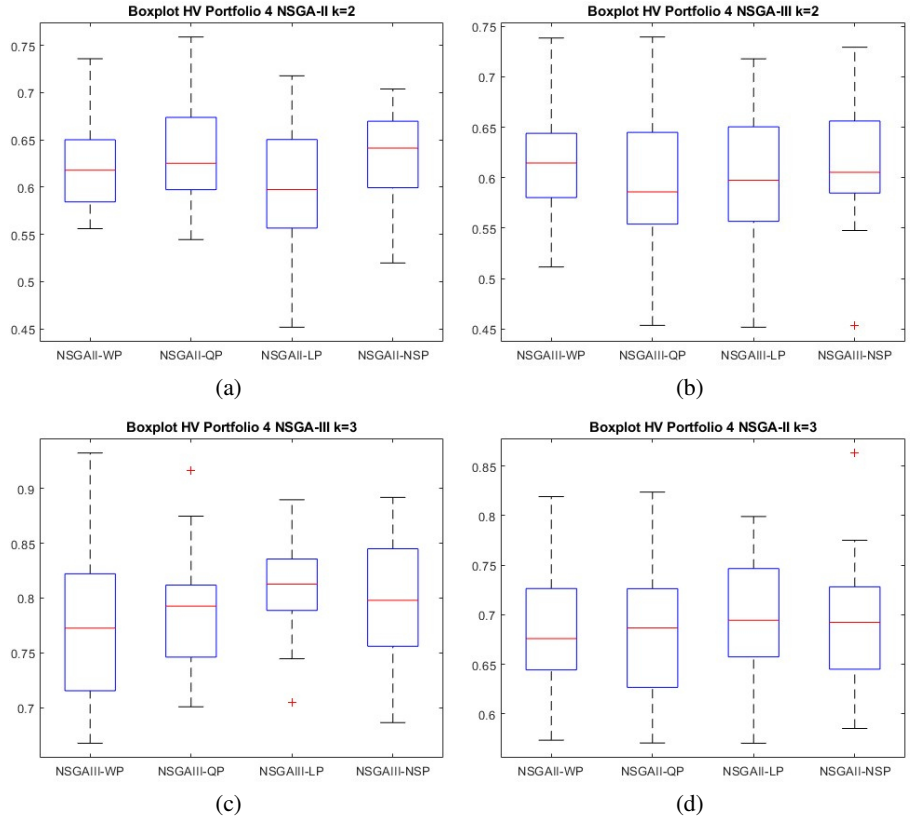


Fig. 2. Boxplots corresponding to HV indicator of Portfolio 4 for two and three objectives.

Notice that Equation 8 only considers equality constraints; thus, inequalities must be transformed. The Augmented Lagrangian Method is presented in Algorithm 3. In this method, the choice of the initial point $\boldsymbol{x}_{k+1}^{s}$ is less critical when using this method.

3 Proposal

As mentioned in Section 2, MOEAs are useful tools in solving CMOPs. However, these algorithms do not always have a penalty strategy to guide them toward feasible solutions. There are different penalty methods available, in this work, three different methods were employed:

The Quadratic Penalty Method, Nonsmooth Penalty Functions and the Augmented Lagrangian Method [23] to work cooperatively with the selected MOEAs (NSGA-II and NSGA-III [10, 16]). In the following, we present the selected penalty strategies.

3.1 Numerical Results

This section is dedicated to observe the impact that penalty strategies have when used in CMOPs, specifically in the Portfolio Optimization Problem.

Table 2. Average value of the performance indicators of the portfolio problem for $n = 5, 10, 20, 30, 40, 50$ with $k = 2$ via NSGA-II and NSGA-III without penalty strategy (WP), with quadratic penalty (QP), with Nonsmooth penalty (NSP) and with Lagrangian penalty (LP).

NSGA-II												
	FR				Δ_p				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.9403	0.9883	**1**	0.9917	**1.9185e-04**	7.5454e-04	7.7572e-04	6.5893e-04	0.5728	0.5500	0.5545	**0.6083**
(std.dev)	0.0259	0.0018	0	0.0069	1.8981e-04	1.6311e-04	1.6315e-04	3.5919e-04	2.1016e-04	2.3343e-04	0.0011	0.0029
Portfolio 2	0.8777	0.9423	**0.9997**	0.9460	4.4319e-04	**3.5964e-04**	4.3592e-04	4.7167e-04	0.3327	**0.3535**	0.3321	0.3489
(std.dev)	0.0610	0.0326	0.0018	0.0396	8.7398e-06	2.3808e-05	2.7188e-05	4.0999e-05	0.0878	0.0745	0.0761	0.0995
Portfolio 3	0.6010	**0.8257**	0.7887	0.7710	0.0020	**0.0018**	**0.0018**	**0.0018**	0.6264	0.6310	**0.6445**	0.6199
(std.dev)	0.0660	0.0536	0.0630	0.0541	4.9617e-05	7.3861e-05	5.8708e-05	6.5815e-05	0.0516	0.0.0536	0.0630	0.0541
Portfolio 4	0.8147	**0.9250**	0.9193	0.9227	9.6001e-04	9.5479e-04	**9.0018e-04**	9.7601e-04	0.6233	**0.6360**	0.6352	0.6290
(std.dev)	0.0630	0.0443	0.0370	0.0451	1.1139e-05	1.0845e-05	1.3419e-05	1.0860e-05	0.0537	0.0649	0.0595	0.0626
Portfolio 5	0.8027	0.9087	**0.9240**	0.9230	0.0018	**0.0017**	0.0018	**0.0017**	0.7206	**0.7413**	0.7387	0.7400
(std.dev)	0.0807	0.0537	0.0368	0.0537	2.2405e-05	1.6720e-05	1.6461e-05	1.6641e-05	0.0530	0.0671	0.0628	0.0639
Portfolio 6	0.8083	0.9170	**0.9243**	0.9163	0.0019	**0.0018**	**0.0018**	**0.0018**	0.6907	0.6848	**0.7018**	0.6838
(std.dev)	0.0659	0.0432	0.0362	0.0415	1.5479e-05	1.8858e-05	1.0395e-05	1.8708e-05	0.0543	0.0516	0.0499	0.0512
NSGA-III												
	FR				Δ_p				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.9743	0.9987	**1**	0.9953	6.6309e-04	**6.0168e-04**	8.5694e-04	0.0013	0.5706	0.5670	0.5551	**0.6528**
(std.dev)	0.0179	0.0035	0	0.0035	1.9198e-04	2.0051e-04	1.3804e-04	3.3193e-04	2.0786e-04	3.3168e-04	5.0220e-04	0.0031
Portfolio 2	0.9490	0.9887	**0.9997**	0.9930	4.6256e-04	**4.3991e-04**	4.7078e-04	6.4490e-04	0.3320	**0.3396**	0.3195	0.3374
(std.dev)	0.0252	0.0063	0.0018	0.0065	7.9075e-06	1.2824e-06	3.5864e-05	2.5302e-05	0.0901	0.0822	0.0989	0.1193
Portfolio 3	0.6853	**0.9283**	0.9033	0.8650	0.0024	0.0023	**0.0022**	0.0023	0.6075	0.6197	0.6283	**0.6286**
(std.dev)	0.1022	0.0385	0.0434	0.0581	6.2940e-05	4.5772e-05	4.0169e-05	4.2092	0.0676	0.0604	0.0713	0.1022
Portfolio 4	0.9010	**0.9627**	0.9623	0.9577	0.0011	0.0011	0.0011	**0.0010**	**0.6183**	0.5958	0.6132	0.6033
(std.dev)	0.0491	0.0215	0.0319	0.0275	9.0502e-06	9.8507e-06	1.9612e-05	1.4121e-05	0.0627	0.0747	0.0632	0.0742
Portfolio 5	0.8717	0.9537	**0.9620**	0.9513	0.0020	**0.0019**	**0.0019**	**0.0019**	0.7249	**0.7341**	0.7320	0.7317
(std.dev)	0.0589	0.0361	0.0277	0.0359	2.0833e-05	2.0766e-05	2.1443e-05	2.0607e-05	0.0604	0.0660	0.0617	0.0668
Portfolio 6	0.8557	0.9457	**0.9517**	0.9460	0.0020	**0.0019**	0.0020	**0.0019**	0.7074	0.6933	**0.7256**	0.6944
(std.dev)	0.0704	0.0332	0.0296	0.0333	2.4282e-05	2.8865e-05	1.6134e-05	2.8940e-05	0.0651	0.0623	0.0472	0.0621

For the numerical experiments, we considered six portfolio problems (for $k = 2$ and $k = 3$, see Equation (3) and Equation (4) respectively), each one related to a different number of assets (5, 10, 20, 30, 40, 50, respectively). We compared the behavior of NSGA-II, NSGA-III, and MOPSO [22] when solving each one of the portfolio problems without penalization (WP) and with different types of penalty strategies (QP, NSP, LP).

For all experiments, we have executed 30 independent runs using 100,000 function evaluations. For the numerical experiments, we used PlatEMO [28]. Table 1 contains the algorithm parameter values used for the experimental setting. The performance indicators Δ_p and $\mathrm{Hypervolume(Hv)}$ [26, 4, 31, 32] are used to measure the penalty strategy effectiveness.

In this work, the real PF used to compute the Δ_p indicator is obtained by theoretically solving the Portfolio Optimization Problem. To compute the Hv indicator, we normalized each objective value of the approximated solution and then set the reference point as $[1,\ 1]$ for two objectives and $[1,\ 1,\ 1]$ for three objectives. We also measure the feasibility rate (FR) of each run. FR is defined as:

$$\mathrm{FR} = \frac{\text{number of feasible individuals}}{\text{number of total individuals}}. \tag{10}$$

Table 3. Average value of the performance indicators of the portfolio problem for $n = 5, 10, 20, 30, 40, 50$ with $k = 3$ via NSGA-II and NSGA-III without penalty strategy (WP), with quadratic penalty (QP), with Nonsmooth penalty (NSP) and with Lagrangian penalty (LP).

NSGA-II								
	FR				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.5386	0.8214	**0.8931**	0.7567	0.5971	0.6329	0.6316	**0.6386**
(std.dev)	0.0320	0.0299	0.0120	0.0824	0.0606	0.0888	0.1225	0.0968
Portfolio 2	0.5070	0.7257	**0.7270**	0.7313	0.5469	**0.5730**	0.5566	0.5618
(std.dev)	0.0426	0.0364	0.0469	0.0450	0.0522	0.0408	0.0368	0.0478
Portfolio 3	0.4437	0.6983	0.6940	**0.7053**	0.7025	**0.7254**	0.7155	0.7179
(std.dev)	0.0491	0.0318	0.0294	0.0487	0.0911	0.0783	0.0982	0.0623
Portfolio 4	0.4227	0.6750	0.6730	**0.6810**	0.6857	0.6815	0.6920	**0.6980**
(std.dev)	0.0498	0.0367	0.0455	0.0370	0.0628	0.0631	0.0623	0.0582
Portfolio 5	0.4150	**0.6770**	0.6757	0.6730	0.6963	0.7007	**0.7065**	0.7047
(std.dev)	0.44	0.0503	0.0398	0.0497	0.0761	0.0701	0.0636	0.0653
Portfolio 6	0.4180	0.6500	**0.6593**	0.6497	0.7272	0.7212	0.7438	**0.7471**
(std.dev)	0.0387	0.0409	0.0486	0.0472	0.0668	0.0598	0.0761	0.0677
NSGA-III								
	FR				**Hv**			
	WP	QP	NSP	LP	WP	QP	NSP	LP
Portfolio 1	0.5994	0.8975	**0.9481**	0.8753	0.7049	0.7291	**0.7471**	0.6936
(std.dev)	0.0399	0.0295	0.0221	0.0905	0.1665	0.1395	0.1521	0.1550
Portfolio 2	0.5642	0.7781	0.7933	**0.7969**	0.6144	0.6213	0.6131	**0.6285**
(std.dev)	0.0473	0.0362	0.0387	0.0633	0.0464	0.0444	0.0346	0.0544
Portfolio 3	0.4772	**0.7283**	0.7083	0.7103	0.7846	0.7916	0.7892	**0.7928**
(std.dev)	0.0531	0.0320	0.0375	0.0460	0.0627	0.0652	0.0481	0.0571
Portfolio 4	0.4150	0.6847	**0.6944**	0.6733	0.7750	0.7873	0.7951	**0.8106**
(std.dev)	0.0488	0.0425	0.0349	0.0386	0.0644	0.0508	0.0532	0.0425
Portfolio 5	0.4008	0.6628	0.6636	**0.6719**	0.7890	**0.8083**	0.7985	0.7962
(std.dev)	0.0436	0.0377	0.0527	0.0376	0.0592	0.0551	0.0499	0.0620
Portfolio 6	0.3861	0.6536	**0.6578**	0.6450	0.7819	0.8016	**0.8194**	0.7946
(std.dev)	0.0370	0.0495	0.0459	0.0461	0.0539	0.0610	0.0607	0.0542

We claim that by using penalty strategies, not only does the feasibility rate improve, but we also improve the performance of the MOEAs when solving the portfolio optimization problem. Table 2 shows the obtained results using NSGA-II and NSGA-III for the portfolio problem of $k = 2$ and Table 3 shows the obtained results using NSGA-II and NSGA-III for the portfolio problem of $k = 3$.

MOPSO algorithm had troubles when solving the selected CMOPs. When the number of assets increases ($n > 10$), the algorithm fails in finding feasible solutions. Figure 1 shows the behavior of NSGA-II with and without penalty strategy on a certain execution for Portfolio 1 and 3.

Observe that more feasible solutions can be obtained by implementing a penalty function. First, recall that the FR indicator measures feasibility. If the indicator value tends to 1, there is a higher prevalence of feasible solutions. Note that in all portfolios, the FR indicator is always higher when a penalty function is applied. Additionally, the WP value is always the smallest, meaning NSGA will always obtain more feasible solutions by incorporating a penalty strategy.

Now, the Δp indicator aims for convergence and distribution; a smaller value indicates higher performance. Analyzing the case of $k = 2$, the first portfolio, we notice that the Δp indicator without applying penalty functions is lower than when a penalty function is applied; in this case, we are considering only five decision variables. Therefore, the standalone NSGA-II is enough to solve the problem.

However, in the remaining five portfolios (more variables), this indicator is always better when some penalty function is applied. Finally, we have the Hv indicator, which measures the volume of the space dominated by a set of solutions in the objective space, so this indicator should tend to 1 when all objectives are normalized.

Note that the Hv indicator is higher in all portfolios when some penalty function is applied. Note that for NSGA-III, we have a similar behavior. In all portfolios, the FR indicator is always higher when some penalty function is applied, and the value of the WP indicator is always the smallest. Also, the penalty versions outbeat the standalone algorithm referring to Δ_p and Hv indicators. Only in Portfolio 3 the higher value of HV is found in WP.

Finally, considering the portfolio problem for $k = 3$ one can notice that, as expected, penalty strategies helped the evolutionary framework and obtained solutions of higher quality. Although the FR value is no longer as good as for $k = 2$, it still is better than the standalone version.

For this case, we only measure the Hv indicator since we do not know the real Pareto front of the problem. Finally, boxplots corresponding to the Portfolio 4 problem for two and three objectives using the Hv indicator are presented in Figure 2. Observe that the MOEA version with a penalty strategy gets better results than the standalone algorithm.

4 Conclusions

In this work, we deal with the Portfolio Optimization Problem; since it can be defined as a CMOP, we are interested in analyzing how well it can be solved by MOEAs considering penalty strategies. Our proposal uses three different penalty functions, each with a specific characteristic. We claim that better performance is expected when a MOEA employs a penalty strategy. Several numerical results support this claim.

Numerical experiments showed that penalty strategies helped the behavior of the evolutionary framework, not only in terms of performance indicators (Δ_p and Hv) but also by obtaining a major number of feasible individuals. Since it is a constrained optimization problem, we aim for feasibility and optimality. Although we have promising results, it is worth noticing that these are preliminary results since some aspects are still pending exploration. For example, we focused here on the most common portfolio optimization problem (2 and 3 objectives), but this problem can be more difficult if more constraints are considered, as the ones used in [2].

Our next steps include studying different types of portfolios and analyzing which penalization strategy is more suitable for these types of MOPs, aiming for theoretical results that back up them.

Acknowledgments. Lourdes Uribe, Uriel Trejo-Ramirez and Yael Andrade-Ibarra acknowledge support from project no. SIP20232208. Oliver Cuate and Victor Cordero acknowledge support from project no. SIP20221947 and IPN-SIP 20231045.

References

1. Awad, M., Abouhawwash, M., Agiza, N. H.: On NSGA-II and NSGA-III in portfolio management. Intelligent Automation and Soft Computing, vol. 32, no. 3, pp. 1893–1904 (2022) doi: 10.32604/iasc.2022.023510
2. Bacanin, N., Tuba, M.: Fireworks algorithm applied to constrained portfolio optimization problem. In: IEEE Congress on Evolutionary Computation, pp. 1242–1249 (2015) doi: 10.1109/CEC.2015.7257031
3. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research, vol. 181, no. 3, pp. 1653–1669 (2007) doi: 10.1016/j.ejor.2006.08.008
4. Bogoya, J. M., Vargas, A., Schütze, O.: The averaged Hausdorff distances in multi-objective optimization: A review. Mathematics Multidisciplinary Digital Publishing Institute, vol. 7, no. 10, pp. 1–35 (2019) doi: 10.3390/math7100894
5. Coello-Coello, C. A., Lamont, G. B., Van-Veldhuizen, D. A.: Evolutionary algorithms for solving multi-objective problems. Springer (2007)
6. Coit, D. W., Smith, A. E., Tate, D. M.: Adaptive penalty methods for genetic optimization of constrained combinatorial problems. Institute for Operations Research and the Management Sciences Journal on Computing, vol. 8, no. 2, pp. 173–182 (1996) doi: 10.1287/ijoc.8.2.173
7. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, vol. 186, no. 2–4, pp. 311–338 (2000) doi: 10.1016/s0045-7825(99)00389-8
8. Deb, K.: Multi-objective optimization using evolutionary algorithms: An Introduction. John Wiley and Sons, Inc (2011) doi: 10.1007/978-0-85729-652-8_1
9. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 577–601 (2014) doi: 10.1109/TEVC.2013.2281535
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197 (2002) doi: 10.1109/4235.996017
11. Deng, G. F., Lin, W. T.: Ant colony optimization for Markowitz mean-variance portfolio model. In: Swarm, Evolutionary, and Memetic Computing: First International Conference on Swarm, Evolutionary, and Memetic Computing, vol. 6466, pp. 238–245 (2010) doi: 10.1007/978-3-642-17563-3_29
12. Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., Goodman, E.: Push and pull search for solving constrained multi-objective optimization problems. Swarm and evolutionary computation, vol. 44, pp. 665–679 (2019) doi: 10.1016/j.swevo.2018.08.017
13. Goldberg, D. E.: Genetic algorithms in search, optimization and machine learning. ACM Digital Library (1989)

14. Golmakani, H. R., Fazel, M.: Constrained portfolio selection using particle swarm optimization. Expert Systems with Applications, vol. 38, no. 7, pp. 8327–8335 (2011) doi: 10.1016/j.eswa.2011.01.020
15. He, Q., Wang, L.: A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation, vol. 186, no. 2, pp. 1407–1422 (2007) doi: 10.1016/j.amc.2006.07.134
16. Jain, H., Deb, K.: An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 602–622 (2014) doi: 10.1109/tevc.2013.2281534
17. Jana, P., Roy, T. K., Mazumder, S. K.: Multi-objective possibilistic model for portfolio selection with transaction cost. Journal of Computational and Applied Mathematics, vol. 228, no. 1, pp. 188–196 (2009) doi: 10.1016/j.cam.2008.09.008
18. Kamal-Abasi, A., Tajudin-Khader, A., Al-Betar, M. A., Naim, S., Makhadmeh, S. N., Alkareem-Alyasseri, Z. A.: Link-based multi-verse optimizer for text documents clustering. Applied Soft Computing, vol. 87, pp. 106002 (2020) doi: 10.1016/j.asoc.2019.106002
19. Knypiński, Ł.: Adaptation of the penalty function method to genetic algorithm in electromagnetic devices designing. The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, vol. 38, no. 4, pp. 1285–1294 (2019) doi: 10.1108/COMPEL-01-2019-0010
20. Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., Gao, Z.: A hybrid feature selection algorithm for gene expression data classification. Neurocomputing, vol. 256, pp. 56–62 (2017) doi: 10.1016/j.neucom.2016.07.080
21. Mezura-Montes, E., Coello-Coello, C. A., Tun-Morales, E. I.: Simple feasibility rules and differential evolution for constrained optimization. In: Mexican International Conference on Artificial Intelligence, pp. 707–716 (2004) doi: 10.1007/978-3-540-24694-7_73
22. Moore, J., Chapman, R.: Application of particle swarm to multiobjective optimization. Department of Computec Science and Software Engineering, Anburn University (1999)
23. Nocedal, J., Wright, S. J.: Numerical optimization. Springer (1999)
24. Ravizza, S., Chen, J., Atkin, J. A. D., Burke, E. K., Stewart, P.: The trade-off between taxi time and fuel consumption in airport ground movement. Public Transport, vol. 5, no. 1–2, pp. 25–40 (2013) doi: 10.1007/s12469-013-0060-1
25. Richardson, J. T., Palmer, M. R., Liepins, G. E., Hilliard, M. R.: Some guidelines for genetic algorithms with penalty functions. In: Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 191–197 (1989)
26. Schutze, O., Esquivel, X., Lara, A., Coello-Coello, C. A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation, vol. 16, no. 4, pp. 504–522 (2012) doi: 10.1109/tevc.2011.2161872
27. Sefiane, S., Benbouziane, M.: Portfolio selection using genetic algorithm. Journal of Applied Finance and Banking, vol. 2, no. 4, pp. 143–154 (2012)
28. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Computational Intelligence Magazine, vol. 12, no. 4, pp. 73–87 (2017) doi: 10.1109/mci.2017.2742868
29. Zhang, K., Du, H., Feldman, M. W.: Maximizing influence in a social network: Improved results using a genetic algorithm. Physica A: Statistical Mechanics and its Applications, vol. 478, pp. 20–30 (2017) doi: 10.1016/j.physa.2017.02.067
30. Zhu, H., Wang, Y., Wang, K., Chen, Y.: Particle swarm optimization (PSO) for the constrained portfolio optimization problem. Expert Systems with Applications, vol. 38, no. 8, pp. 10161–10169 (2011) doi: 10.1016/j.eswa.2011.02.075

31. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - a comparative case study. In: Parallel Problem Solving from Nature, pp. 292–301 (1998) doi: 10.1007/bfb0056872
32. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., da Fonseca, V. G.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation, vol. 7, no. 2, pp. 117–132 (2003) doi: 10.1109/tevc.2003.810758

ISSN 1870-4069

Self-Supervised Learning with Legal-Related Corpus: Customizing a Language Model with Synthetic Data

Philippe Prince-Tritto[1], Hiram Ponce[2]

[1] Universidad Panamericana,
Facultad de Derecho,
Mexico

[2] Universidad Panamericana,
Facultad de Ingeniería,
Mexico

{pprince, hponce}@up.edu.mx

Abstract. This paper explores the development of a customized text generation system using pre-trained language models, specifically aimed at knowledge workers such as lawyers and personal data protection specialists. Our approach minimizes human intervention in the labeling process for fine-tuning. To this end, we automate data collection and filter the data through GPT-3.5 and BERT-based heuristics. Human expertise is only leveraged in design and oversight, ensuring the system's ability to provide accurate and relevant information. We also developed an annotation tool to complete our training set, utilizing text generation which required a low level of human supervision. This paper repurposes the Prompt Generation Network architecture to create a chatbot in Spanish language that can address queries related to personal data protection. Our results showed encouraging progress towards automating the annotation of a dataset for fine-tuning with little human intervention, although opportunities for improvement remain. Ultimately, our research offers a blueprint for the creation of a chatbot using a fine-tuned language model with minimal human intervention, demonstrating the potential of these models for practical applications.

Keywords: Self-supervised learning, legal language processing, fine-tuning, large language model.

1 Introduction

The rapid advancement in the field of Machine Learning (ML) and natural language processing (NLP) has led to the development of increasingly sophisticated language models, capable of analyzing and predicting human-like text. These models, such as GPT [9] and LLaMa [10], hold immense potential for a wide range of applications, from personalized assistance to professional tools for knowledge workers.

However, harnessing the power of these pre-trained models often requires a fine-tuning process that can be time-consuming and resource-intensive, particularly when it comes to the labeling of training data. In this paper, we explore the possibility of utilizing pre-trained language models to create a customized text generation system that caters to the specific needs of knowledge workers, such as lawyers and specialists

in personal data protection. Our objective is to minimize human intervention by automating the labeling process using pre-trained models like GPT-3.5 and BERT for data filtering and question identification. Human expertise is reserved only for design and oversight tasks that require nuanced judgment. Upon existing research on the efficiency of pre-trained language models, prompt-learning architectures, and the scalability of text generation from prompts, we can claim that these techniques can be combined to develop a robust, lightly supervised annotation tool.

Some authors have focused primarily on the effectiveness of pre-trained language models to be refined to recognize and respond to specific prompts [11]. The proposed architectures are intended to be applied in the pre-training phase to improve the efficiency of fine-tuning. For this paper, we are inspired by the Prompt Generation Network (PGN) architecture [6] for Prompt-learning.

PGN consists in generating input-dependent prompts by sampling from a learned library of tokens. It should be noted that for these authors, the task-specific data are in the pixel space, while in our case it is text. The scalability of text generation from prompts could provide immense potential for the development of robust annotation tools that require fairly low human supervision. Transformers systems have the ability to generate their own tags and patterns for learning how to learn.

They are able to do this in a way that is more efficient and less costly than human-generated annotation [12]. This can also alleviate the shortage of labeled data, which is an obstacle to better performance, and thus enrich the model representations [5]. It should be understood that today, the main bottleneck when training a language model, or any supervised learning, is data labeling. However, some [3] have shown that the use of pre-trained models can surpass human intervention in this repetitive, mind-numbing task for the operator, and with little added value from a humanistic perspective.

In this study, the role of human intervention was primarily in the design and oversight of the data collection, with a thorough ETL process, and model fine-tuning processes. The web scraping process was automated but designed by human engineers. The initial dataset of articles was filtered using a GPT-3.5 model, reducing the need for human curation. However, human judgment was applied in the design of the BERT-based heuristic for question identification and in the choice of hyperparameters for model fine-tuning. The aim was to minimize human involvement in the routine tasks of data labeling and curation, while still leveraging human expertise for tasks that required nuanced judgment.

We aim to demonstrate that the availability of language models such as GPT or LLaMa opens up a potential for customizing ML models for knowledge workers. The problem is thus the following: How can we use pre-trained language models to produce a text generation system that relies on specific knowledge, with the least amount of human intervention with respect to the labeling of the model's fine tuning data?

To answer this, we developed a Spanish chatbot capable of addressing laypeople's queries related to personal data protection. Fine tuning a specific LLM requires data labelized as prompt and completions. For the completions, we implemented our approach by scraping specific web databases comprising privacy-related newspaper articles, cleaning the data, and separating it into target responses.

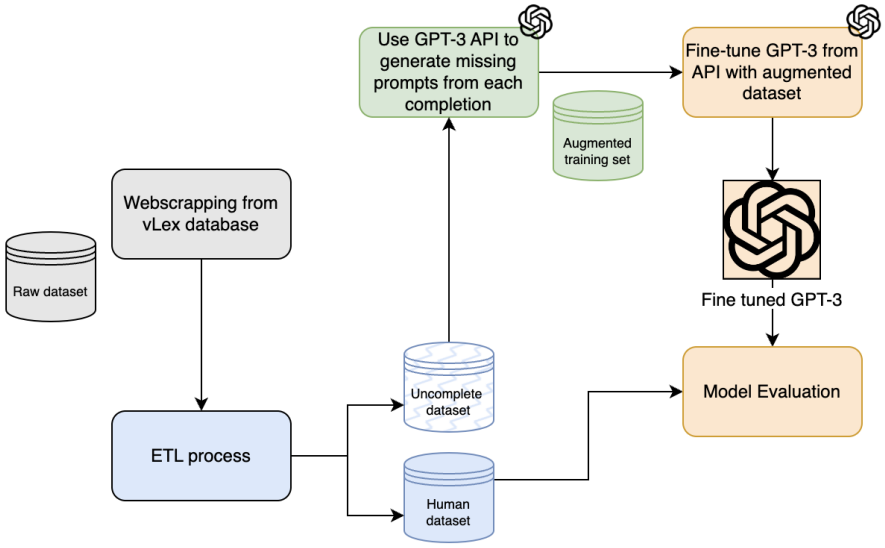


Fig. 1. Flowchart diagram for the self-supervised proposal.

These responses needed questions: the corresponding 'prompt'. GPT-3.5 was utilized to generate the prompt for each given completion for fine-tuning the LLM without expert intervention. The evaluation involved (1) a fully human-curated dataset, derived from the web scraping step, and (2) a partially synthetic dataset, where prompts were generated by GPT-3.5 and completion was taken from the scraped data.

Each dataset provided a basis for comparing the responses generated by the fine-tuned model to the target output. The rest of the paper is organized as follows. In Section 2, we present the working experiment of creating a Spanish-language chatbot capable of addressing questions and concerns related to personal data protection, from an non-annotated dataset.

We detail the experimentation's execution, and then examine the efficacy of this chatbot in Section 3. After demonstrating the potential for customized ML models to assist knowledge workers in their professional endeavors, we conclude in Section 4 on the use of the advancements in NLP without placing undue burden on human resources.

2 Materials and Methods

We propose to fine-tune a large language model to create an agent for lay people to resolve their doubts about the requirements of personal data protection in Spanish language and in Mexico. The focus of the experiment is on the dataset used for the model fine tuning that underlies the chatbot. We exploit a pre-trained model to generate sound training data. Following [7], we go one step further incorporating some synthetic data in the training database to improve its performance, as shown in Figure 1.

The database consists of non-annotated legal news articles from vLex, obtained with webscrapping techniques. We assume that the authors of these news articles are answering a specific question about personal data. What is missing is a formulation of the question they answer. It is this question, the prompt, that we propose to generate with a pre-trained language model.

We then reuse this synthetic data to fine-tune this same language model. Our agent is a chatbot that generates through it's interface text fragments on the specific topic of personal data protection in Spanish. The experiment takes the form of a proof-of-concept to validate the hypothesis.

2.1 ETL Process

Description of the Initial Dataset. The first task is to retrieve data related to privacy. In order to work with real data and replicate the process used in a company wishing to exploit its own data, we downloaded via a webscrapping process the data from the vLex platform, searching for the exact term "Protección de datos personales" (Personal data protection) and filtering by type of documents.

To carry out our experiment, we limited ourselves to press articles on the subject. As the articles were classified by relevance, we used this classification for webscrapping and retrieved the first 4091 results. To remove noise from the retrieved data, we filtered the results using a binary classification with GPT-3.5, based on the title of the article. The prompt used for the classification task was:

```
Olvida todas las instrucciones anteriores. Eres un
clasificador de noticias en materia de privacidad en México.
Basándose en el análisis del título, tu tarea consiste en
responder '1' si es probable que el artículo trate de la
protección de datos personales, y '0' en caso contrario. El
título es: 'title'.
```

The English translation of the prompt is:

```
Forget all previous instructions. You are a news classifier
focused on privacy matters in Mexico. Based on the title
analysis, your task is to answer '1' if it's likely that the
article is about personal data protection, and '0' otherwise.
The title is: '{title}'.
```

This allowed us to determine that the number of articles directly related to privacy was 737. This simple step using a heuristic based on an existing language model allowed us to keep only relevant data at minimal cost. It seemed reasonable to not webscrap the whole vLex database, since we did not have a lot of relevant items after reaching a certain point. We determined this point with a plot that shows a smoothed rate of change in the number of privacy related articles.

When the smoothed rate of change approaches zero, it indicates that there is no significant increase in the number of privacy-related documents anymore, as shown in Figure 2. The articles were then separated into paragraphs, with each paragraph representing a human-data sample.

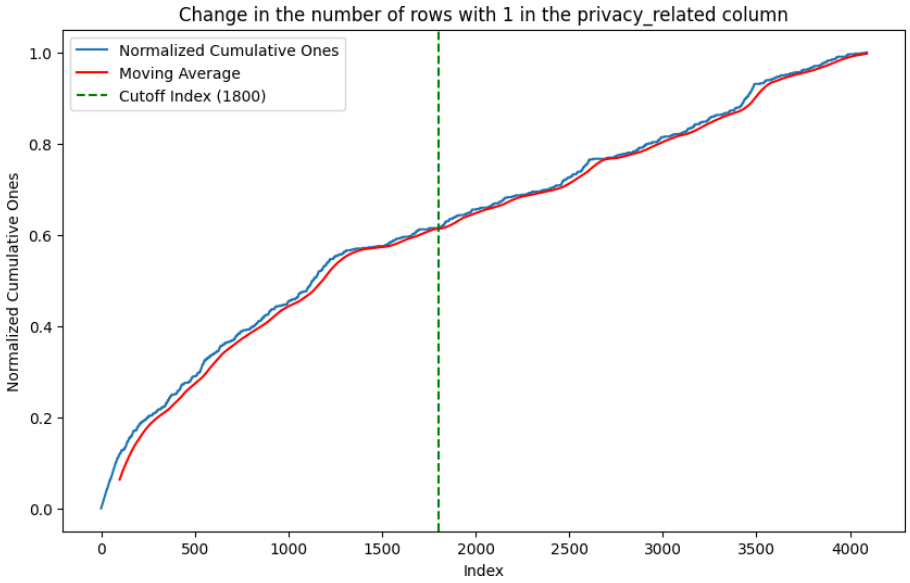


Fig. 2. Smooth rate of change in the privacy-related document. The cutoff index is percentile-based so that the threshold is equal to 0.00003.

Each sample was normalized, removing the logical connectors at the beginning of the paragraph to make it look like an answer to a question. As many irrelevant paragraphs as possible were removed from the dataset, for example when they began with certain cues that announced they were advertising paragraphs. We also utilized spaCy to eliminate samples that began with a proper name, as this information was not pertinent to a general chatbot focusing on personal data.

Most paragraphs of this nature lacked substance but instead provided details such as the author's identity and a summary of their professional background. To build a base of Full Human data, we then identified the questions in our dataset, storing them as 'prompt'. The following paragraphs were stored as 'completion'. In order to determine which paragraphs could be used as answers, we performed a Similarity-based heuristic with BERT.

Due to the small number of questions identified in our dataset, we also assumed that the shortest paragraphs were titles and we based our heuristic on the fact that the first paragraphs following a title give a short answer to it. We therefore stored as 'prompt' headlines of 15 words or and concatenated "What can you tell me about" with the headline. We then repeated the Similarity-based heuristic with BERT to store the following paragraphs as 'completion'.

From the 737 privacy related articles, the above method allowed us to get to 4177 human-data samples that constituted the completions to the synthetic prompts we would further generate, and 545 tuples of Full Human-data prompts and completions, as shown in Figure 3. Full Human dataset would further be used for testing purposes only. After estimating the cost of fine-tuning an OpenAI davinci model, including the generation of synthetic prompts, we moved on to generating the training set with synthetic data.

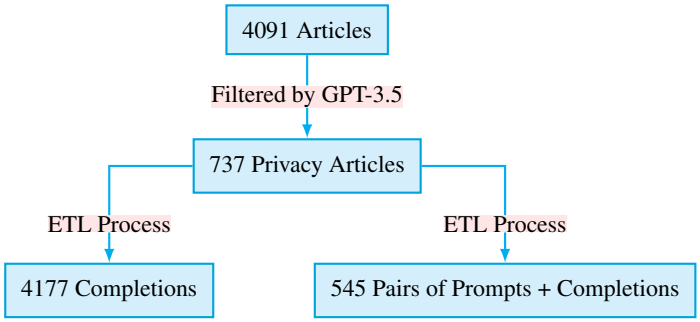


Fig. 3. From webscraping to filtered dataset for synthetic data generation for fine-tuning and full human validation set.

Prompt Generation for Synthetic Data and Final Dataset Summary On a sample of 10 completions, we performed a grid search to determine the hyperparameters of the davinci model of GPT-3.5 as shown in Table 1. As for the Partially synthetic training and testing data, we designed a prompt to guide the language model in its response. It goes as follows:

```
Eres experto en protección de datos personales en México.
Genera un prompt conciso y corto en idioma Español que podría
ser el mejor candidato para ser contestado por el siguiente
texto en materia de protección de datos personales en México:
``{row[``completion'']}''
```

The English translation of the prompt is:

```
You are an expert in personal data protection in Mexico.
Generate a concise and short prompt in Spanish language that
could be the best candidate to be answered by the following
text on the subject of personal data protection in Mexico:
``{row[``completion'']}''
```

In other words, the Partially Synthetic dataset is composed of tuples of synthetic-prompt and human-completion samples. Generating the prompts corresponding to each of the completions resulted in a dataset of 4177 tuples of synthetic-prompt and human-completion samples. This dataset was separated such that 16% constituted the Partially Synthetic test set for the fine tuning of the model, and the remainder the Partially Synthetic training set. Some examples are given in Table 2.

As described above (see Figure 3), the Full Human test set to validate the fine-tuned model is composed of 545 data points. This dataset is composed of tuples of human-prompt and human-completion samples. Some examples are given in Table 3. We represented the datasets summary in Table 4. An essential aspect of our methodology is the validation of the generated data. While the synthetic prompts are generated by a fine-tuned language model, their pairing with human-generated completions ensures a level of quality and relevance.

Table 1. Hyperparameter Grid search for Prompt Generation (Results in bold).

Temperature	Top-P	Frequency Penalty	Presence Penalty
0.5	0.5	0.5	**0.5**
0.8	**0.8**	0.8	0.8
1.0	1.0	**1.0**	1.0

These pairs undergo a systematic filtering process, as outlined in Section 2.1, to remove any outliers or irrelevant entries. Similarly, the Full Human dataset is derived from vetted, privacy-related articles, adding another layer of quality control. No manual corrections are applied to the data; instead, we rely on the rigor of our automated processes and the fine-tuning and evaluation metrics to ensure data integrity.

We further used the synthetic-prompt from the Partially Synthetic test set and human-prompt from the Full Human test set to further validate the fine-tuned model, generating completion from those prompts and comparing it with the human-completion of the Partially Synthetic test set and the Full Human test set.

Model Fine-Tuning. For the fine-tuning process, we exclusively utilized the Partially Synthetic dataset. This contextual detail is pivotal for interpreting the subsequent performance evaluation of the fine-tuned model.

We used Weight and Bias for monitoring and the OpenAI API to fine tune the davinci GPT model. The hyperparameters chosen for this step were not subject to a grid search because of the cost that this could represent. For the hyperparameters, we used a Batch size of 64, which is the high limit of what is commonly practiced, a learning rate of 0.01 and 4 epochs, in order to avoid overfitting while preserving training costs.

Finally, since all tasks are equally important in the task of our language model, we set a prompt loss weight of 1.0. The OpenAI API for fine-tuning allowed us to measure the loss (for assessing if the model is learning and fitting the training data well and performs well with unseen examples) and token accuracy (for assessing if the model predicts the correct token) for both the training and validation sets.

It is worth noting that when fine-tuning the model, all layers are retrained, since fine-tuning is a process that adjusts all the weights and biases in the model, across all layers [4]. The purpose of fine-tuning is to adapt a pre-trained model, which was originally trained on a large, diverse dataset, to perform well on a specific task or to better match a narrower dataset.

2.2 Performance Evaluation

To evaluate the model performance, we use two validation sets of questions that the model has not seen before. We compare the agent's responses to the correct answers. Two tests are performed. The first one, where both the prompt and the answer are natural data (Human Set Validation, see Table 6). The other one, where the prompt is synthetic data and the answer is natural data (Synthetic Set Validation, see Table 6). We then proceed both to a quantitative and a qualitative assessment.

Quantitative Assessment. We are unable to calculate perplexity for the generated texts, as this would require knowing the exact probabilities that the model assigns to each word, which is not available with GPT-3.5.

Table 2. Example of synthetic prompts and human completions from the Partially Synthetic Dataset.

Prompt	Completion
¿Cómo se puede prevenir el trashing en México?	Este delito se conoce como trashing y consiste en que los delincuentes obtienen información privada como estados de cuenta, copias de identificaciones oficiales, recibos, documentos, directorios e incluso contraseñas que el usuario ha enviado a la basura o a la papelera de reciclaje de su equipo. Existe tanto de manera física como digital.
¿Qué implicaciones tiene el RFC para la protección de datos personales en México?	La idea de que todos los mayores de 18 años en México cuenten con su Registro Federal de Contribuyentes (RFC) es mucho más que la simple búsqueda de un control fiscal de las autoridades tributarias sobre los ciudadanos. Obligar a todos a tener un RFC forma parte de los esfuerzos del Gobierno federal de construir una matrix para la vigilancia masiva e indiscriminada, en línea con el registro digital de ciudadanos que impulsa la Secretaría de Gobernación y de la construcción de un padrón nacional de datos biométricos de usuarios de telefonía móvil.
¿Qué medidas propone Lorenzo Córdova para garantizar la protección de datos personales en México?	En tanto, Lorenzo Córdova se mostró en la disposición de avanzar en este tema, siempre y cuando no se vulnere el derecho a la protección de datos personales de los ciudadanos, así como que no se viole ninguna normativa que rija a la Secretaría de Gobernación y al Instituto Nacional Electoral.
¿Cuáles son las consecuencias de la omisión cometida por Banamex en materia de protección de datos personales?	De acuerdo con el IFAI, Banamex contravino los principios de consentimiento de calidad y responsabilidad; mantuvo datos inexactos del titular y no hizo la rectificación o cancelación de los mismos, no obstante que resultaba legalmente procedente, por lo que se configuró una omisión que hizo nugatorio el derecho ejercido por el particular.

However, we need to assess the LLM's responses. To this end, it is necessary to determine if the output is correct. We do that using simple unique word overlap, tokenizing both the target and the generated text into individual words and then comparing these sets of words. We calculate similarity scores but do not establish a threshold for determining if a generated text is correct or not.

In other words, True Positives (TP) are the words in the generated text that are also in the target text, False Positives (FP) are the words in the generated text that are not in the target text, and False Negatives (FN) are the words in the target text that are not in the generated text. From there, we were able to compute set-based Precision, Recall and F1 score, as well as ROUGE scores:

– **Precision.** is computed as in Equation 1:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

Table 3. Example of human prompts and completions from the Full Human Dataset.

Prompt	Completion
¿Qué información no comparte WhatsApp con Facebook?	En la política de privacidad, WhatsApp asegura que limita significativamente la información que comparte con Facebook. WhatsApp señala que no comparte: No mantiene registros de con quién hablas o envías mensajes.
¿Qué me puedes decir sobre 'Drones: Opinion 01/2015 on Privacy and Data Protection Issues relating to the Utilisation of Drones'?	Los esfuerzos por garantizar el derecho a la protección de datos personales tienen la finalidad de otorgar a los usuarios de los medios digitales mayor control sobre el uso de su información en manos de organizaciones, sujetos obligados y particulares. En la actualidad, la protección de datos en nuestro país cuenta con un marco jurídico sólido para garantizar su ejercicio.
¿Cuál es el papel de los ciudadanos en la protección de sus datos?	Deben estar conscientes de que su información personal vale y que ante las nuevas tecnologías de la información debemos tener mucho cuidado con los datos que proporcionamos, con las medidas de seguridad de nuestros dispositivos electrónicos. En el uso de la redes sociales es fundamental revisar las declaraciones y los avisos de privacidad.
¿Qué me puedes decir sobre 'Aviso de privacidad'?	El aviso de privacidad garantiza la salvaguarda del derecho de autodeterminación informativa reconocido en las normatividades de protección de datos personales en México.

– **Recall:** Also known as Sensitivity, is computed as in Equation 2:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

– **F1 score:** As shown in Equation 3, is the weighted average of Precision and Recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3}$$

– **ROUGE score:** (Recall-Oriented Understudy for Gisting Evaluation) is a set of metrics[1] for evaluating automatic summarizing of texts as well as machine translation. ROUGE-1 and ROUGE-2 are computed as in Equations 4, 5 and 6, considering the overlap of 1-grams and 2-grams.

ROUGE accounts for the frequency of each word, meaning that duplicate words in both the generated text and the target text are considered. This lead to different Precision and Recall values compared to the aforementioned set-based approach, so it can give a broader scope for evaluation:

Table 4. Datasets used in the self-supervised learning experiment.

Dataset	Train	Test
Partially Synthetic	3480	697
Full Human	0	545

$$\text{ROUGE-N Precision} = \frac{\text{Number of overlapping N-grams}}{\text{Total N-grams in the generated text}}, \tag{4}$$

$$\text{ROUGE-N Recall} = \frac{\text{Number of overlapping N-grams}}{\text{Total N-grams in the target text}}, \tag{5}$$

$$\text{ROUGE-N F1 Score} = 2 \times \frac{\text{ROUGE-N Precision} \times \text{ROUGE-N Recall}}{\text{ROUGE-N Precision} + \text{ROUGE-N Recall}}, \tag{6}$$

where N is the length of the n-gram (e.g., for ROUGE-1, $N = 1$ and the n-grams are individual words; for ROUGE-2, $N = 2$ and the n-grams are two consecutive words, etc.). On the other hand, ROUGE-L considers the Longest Common Subsequence (LCS) between the generated and target texts as shown in Equations 7, 8 and 9. The LCS is a sequence of words that appear in the same order in both texts, although not necessarily consecutively:

$$\text{ROUGE-L Precision} = \frac{\text{Length of LCS}}{\text{Total number of words in the generated text}}, \tag{7}$$

$$\text{ROUGE-L Recall} = \frac{\text{Length of LCS}}{\text{Total number of words in the target text}}, \tag{8}$$

$$\text{ROUGE-L F1 Score} = 2 \times \frac{\text{ROUGE-L Precision} \times \text{ROUGE-L Recall}}{\text{ROUGE-L Precision} + \text{ROUGE-L Recall}}. \tag{9}$$

We calculated ROUGE scores with the ‘rouge.get_scores()‘ function from the ‘rouge‘ Python package. The function returns the F1 scores directly for ROUGE-1, ROUGE-2, and ROUGE-L. However, this function calculates these metrics internally in a similar manner to the formulas mentioned above.

These metrics collectively offer a multi-faceted view of how well the generated text matches the target text, although they do not measure the correctness of the generated text in a semantic or qualitative sense. We also computed BLEU and METEOR scores (overlap between the generated summary and a reference summary) to further validate the answers.

– **BLEU:** (Bilingual Evaluation Understudy) score [8] is a metric that has been developed to evaluate machine translation systems. It compares the n-grams in the generated output and the target, and gives scores between 0 and 1, where 1 is the perfect score as in Equation 10:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \cdot \log(p_n)\right), \tag{10}$$

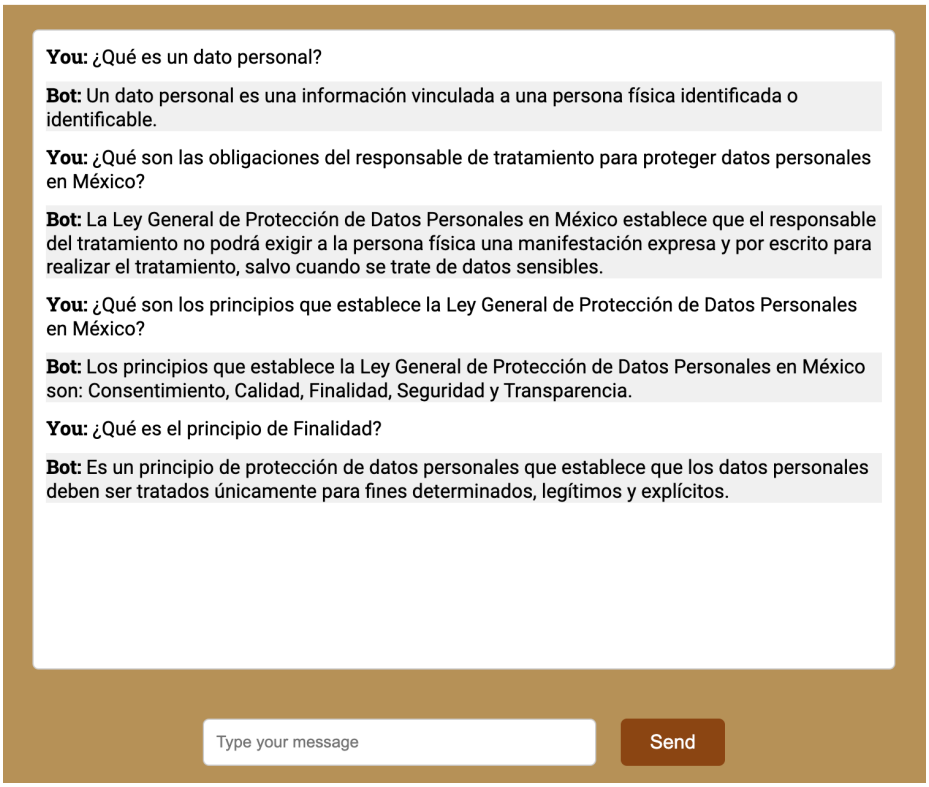


Fig. 4. Chatbot interface showing some prompts and responses.

where:
p_n = is the precision for n-grams.
w_n = is the weight for each n-gram with $w_n = 1/N$.
N = is the maximum order of n-grams used.
BP = is the brevity penalty, calculated as:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r, \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r, \end{cases} \tag{11}$$

where c is the length of the candidate translation and r is the effective reference corpus length.

– **METEOR:** (Metric for Evaluation of Translation with Explicit ORdering) is a metric[2] that has been developed to overcome some of the limitations of metrics such as BLEU. It gives scores between 0 and 1, where 1 is the perfect score. The overall METEOR score is then calculated as in Equation 12:

$$\text{Score} = (1 - \text{Penalty}) \cdot \text{Fmean}, \tag{12}$$

where Penalty is calculated based on the number of chunks (c) and total number of matched unigrams (m) as Penalty $= 0.5 \cdot (\text{c}/\text{m})^3$, and Fmean is the harmonic mean of Precision (P) and Recall (R), with a parameter α set to 0.9 to weight recall more heavily so that:

$$\text{Fmean} = \frac{P \cdot R}{\alpha \cdot P + (1 - \alpha) \cdot R}. \tag{13}$$

Qualitative Assessment. To carry out a qualitative evaluation of the large language model's output, a human evaluator expert in data privacy rated the generated texts based on a defined set of criteria designed to capture important aspects of text quality that are relevant to the evaluation of text generation systems. We assigned weights to each criterion based on their relative importance. The weights w_n reflect the priorities and preferences of the evaluation process.

The evaluation criteria includes: Relevance ($w_1 = 0.2$), which assesses if the generated text aligns with the topic of personal data protection; accuracy ($w_2 = 0.3$), which scrutinizes the correctness and up-to-date nature of the information in the generated text; understandability ($w_3 = 0.15$), which examines if the generated text is easily comprehensible by the target audience; completeness ($w_4 = 0.2$), which measures if the generated text covers all the relevant aspects of the subject; objectivity ($w_5 = 0.1$), which checks for the impartiality and balanced presentation of information; and structure and coherence ($w_6 = 0.05$), which evaluates if the generated text is logically consistent and well-structured.

For each criterion, we compute the average rating from the evaluators, ranging from 0 to 100. We denoted these average ratings as (r_1), (r_2), (r_3), (r_4), (r_5), and (r_6), respectively. We then calculate the Qualitative Score across all evaluators for each criterion, considering the weighted importance, as shown in Equation 14:

$$\text{Qualitative Score} = \frac{\sum_{i=1}^{6} w_i \cdot r_i}{\sum_{i=1}^{6} w_i}. \tag{14}$$

The qualitative assessment was performed on 100 random samples: 50 from the Human Set Validation and 50 from the Synthetic Set Validation.

2.3 Chatbot Implementation

The chatbot implementation makes use of the Flask web framework and the OpenAI's GPT models, leveraging the OpenAI API for conversational responses.

Table 5. Fine Tuning Results.

Metric	Value
Training Loss	0.469
Training Token Accuracy	0.682
Validation Loss	0.729
Validation Token Accuracy	0.676

Two routes are defined - '/' and '/chat'. The first route displays the HTML front end for the chatbot while the second route performs the chatbot processing.

```
@app.route('/')
def index():
return render_template('07_chatbot_front.html')

@app.route('/chat', methods=['POST'])
def chat():
user_message = request.json['message']
```

When the /chat route is accessed, it retrieves the user's message from the JSON payload of the POST request. This message is then used as a prompt to generate a completion from the OpenAI model, with the following hyperparameters:

```
max_tokens=200,
temperature=0.5,
top_p=0.8,
frequency_penalty=1.0,
presence_penalty=0.5,
stop=[''\n'']
```

After receiving a completion, it is processed to remove leading and trailing white space and replacing occurrences of "-¿" (markdown formatting used by our fine-tuned model for answering). The processed completion is then returned to the user interface as a JSON payload, and ploted in an HTML page as shown in Figure 4. The chatbot can be viewed by running the Flask application and opening the specified URL in a web browser, typically 'localhost' with the assigned port number '5000'.

3 Results and Discussion

In this section, we present the results of our experiment, which aimed to fine-tune a large language model for answering personal data protection questions in Spanish language and Mexico. We also discuss the implications of these results in terms of the performance of the chatbot agent.

3.1 Results

The overall results for our fine-tuned model are presented in Table 5. The performance of the chatbot agent was first evaluated using a set of validation questions not seen during training.

The agent's responses were compared to the correct answers, and various evaluation metrics were computed, including precision, recall, F1 score, BLEU score, ROUGE scores and METEOR score. The results are summarized in Table 6. We then carried out the manual human evaluation for qualitative scoring. The results are summarized in Table 7. We also found that the length of generated texts was on average 323% greater when human prompts were presented (178 words) to the fine-tuned model, than when synthetically generated prompts were presented (42 words).

3.2 Discussion

Presenting these separate evaluations in Tables 6 and 7 between the Fill Human and Partially Synthetic sets highlights how well the fine-tuned model performs in different settings—responding to human-generated prompts and synthetic prompts.

We recognize that the Human and Synthetic sets are inherently different, and this is explicitly by design. The Partially Synthetic set is created for the primary purpose of fine-tuning, while the Full Human set serves as a more naturalistic ground truth for performance validation. Therefore this comparison is not intended to show that one is better than the other.

Instead, it offers a multi-faceted evaluation of the model's capabilities. These separate evaluations provide a comprehensive understanding of the model's performance. Based on the metrics presented in Table 5, the model appears to be performing reasonably well, with relatively low training loss (0.469 and 0.729) and moderate token accuracy (0.682 and 0.676) on both training and validation datasets.

However, it's important to consider the specific requirements of personal data protection contents generation. The results presented in Table 6 could be interpreted as a relatively low performance of the chatbot agent in answering questions about personal data protection, suggesting that there is considerable room for improvement in the chatbot's ability to accurately answer questions on this topic. The low ROUGE scores, especially the ROUGE-2 score, also indicate that the generated answers do not closely match the reference summaries.

This could be due to several factors, such as the quality of the training dataset or the limitations of the fine-tuning process. Additionally, the synthetic data generation process may have introduced noise or biases into the training data, which could have negatively impacted the performance of the chatbot. However, it should be noted that for creative or low constrained tasks, such as text generation, it's difficult to assess the quality of an output via a quantitative metric.

Precision, Recall, F1, ROUGE, BLEU and METEOR scores may work as general indicators, but could not be very informative. In these cases, good output can vary enormously, and output that doesn't exactly match the target can still be considered good quality. In addition, these scores are similarity measures based on the presence of common unigrams, bigrams, etc., in the generated output and the target.

They do not capture the semantics or meaning of the output. For example, an output that uses synonyms of words in the target might be semantically very similar to the target, but would have a low ROUGE, BLEU or METEOR score. An important aspect of our findings resides in the analysis of the textual output derived from human-generated prompts and synthetic-generated prompts.

Table 6. Performance evaluation metrics of completions for the fine-tuned model.

Metric	Human Set	Synthetic Set
Average BLEU score	0.0028	**0.0236**
Average Precision	0.1792	**0.3425**
Average Recall	0.2053	**0.3425**
Average F1 Score	0.1521	**0.3350**
Average ROUGE-1 F-score	0.1179	**0.2045**
Average ROUGE-2 F-score	0.0151	**0.0421**
Average ROUGE-L F-score	0.0949	**0.1506**
Average ROUGE-1 Precision	0.1597	**0.2124**
Average ROUGE-2 Precision	0.0220	**0.0444**
Average ROUGE-L Precision	0.1308	**0.1570**
Average ROUGE-1 Recall	0.1431	**0.2124**
Average ROUGE-2 Recall	0.0206	**0.0449**
Average ROUGE-L Recall	0.1167	**0.1564**
Average METEOR score	0.1028	**0.1797**

Our findings reveal a noteworthy trend: the model performs significantly better on synthetic prompts compared to human-generated prompts across multiple evaluation metrics. While it may be tempting to attribute this solely to the model being fine-tuned on synthetic data, it is essential to recognize that these results offer valuable insights into the general interplay between synthetic and human-generated data in natural language processing tasks.

The superior performance with synthetic prompts illuminates possible advantages in their structural and stylistic attributes that make them more conducive for machine interpretation and response generation. This highlights a broader question about the efficacy and limitations of machine learning models in simulating human-like conversational abilities.

It also raises the issue of whether the model's current configuration is sufficiently robust to handle the nuances and complexities inherent in human language. These insights serve to enrich the ongoing discourse on the balance between training data types and model performance, and provide a compelling avenue for future research.

In terms of the qualitative assessment, as presented in Table 7, the generated outputs from synthetic prompts were scored higher in all evaluation criteria, with the overall Quality Score being 55.1 compared to 42.9 for responses generated from human prompts. This suggests that the model performed better when dealing with prompts generated synthetically, indicating a successful transfer of learning.

However, it was noted that the scores for completeness were lower for outputs generated from synthetic prompts, maybe due to the human prompt structure. In addition, the length of generated texts was much longer when human prompts were used, suggesting that synthetic prompts likely lead to more concise responses.

Table 7. Manual human evaluation of completions for qualitative scoring.

	Relevance	Accuracy	Understandability	Completeness	Objectivity	Structure and Coherence	Quality Score
Generated from human prompt	53.0	29.2	59.5	39.1	37.8	**61.3**	42.9
Generated from synthetic prompt	**75.8**	**44.8**	**86.3**	**32.0**	**51.0**	52.9	**55.7**

Further work could investigate if this pattern holds for different domains or languages. There exist several potential avenues for future research aimed at enhancing the efficacy of the chatbot agent. One avenue involves the refinement of the synthetic data generation procedure to produce prompts of superior quality for the training dataset. Another approach entails the inclusion of supplementary sources of training data, coupled with continued efforts to augment the quality and quantity of the training data through the ETL process.

It is important to acknowledge that our study utilized a dataset of relatively modest size, and thus, efforts should be made to enhance its size and diversity, particularly considering that solely news articles were employed for a chatbot that had a legal-related task. Moreover, the qualitative evaluation could be enriched by engaging multiple experts, thereby facilitating a more comprehensive and unbiased assessment. Lastly, further investigation into distinct fine-tuning strategies, hyperparameter optimization, and model architectures has the potential to yield advancements in the chatbot's performance.

4 Conclusions

The chatbot agent demonstrated a limited ability to accurately answer questions about personal data protection in the Spanish language and in Mexico. However, the proposed method for fine-tuning a large language model, specifically for answering personal data protection questions in Spanish, yielded encouraging results. It confirms that a combination of real and synthetic data for fine-tuning can indeed lead to coherent generation of domain-specific text.

In this work, we validated that automation of the annotation of a dataset for fine-tuning is possible with minimal human intervention, primarily focused on design and oversight tasks. It would be necessary to repeat the experiment with a more substantial and diverse dataset, not just legal journalism data, and a larger number of epochs. Our method provides a blueprint for the creation of a chatbot by using fine-tuned language models with few human intervention.

We underlined the potential of these models in practical applications, such as a data protection chatbot that can provide understandable and accurate information to lay users. On a broader scale, this experimental approach highlights how machine learning models can be adapted to specific tasks or domains with the help of fine-tuning strategies, even when a substantial amount of specific task-related training data is not available. It also provides insights for future research on the specifics of fine-tuning these models, which will be increasingly relevant as applications of large language models continue to expand.

One limitation inherent to our methodology was the exclusive utilization of synthetic prompts for fine-tuning the model. Future research endeavors could potentially employ a balanced blend of Human and Synthetic prompts for fine-tuning to engender a model with more robust generalizability across different data domains.

For future work, we will also consider improving the method by integrating a more comprehensive dataset that includes more diversity in terms of topics, formats, and writing styles. In particular, incorporating legal texts, regulatory guidelines, and court case summaries related to personal data protection could enhance the model's understanding of this specific field.

As for evaluation, we could employ a more robust qualitative assessment, involving a larger panel of domain experts, to better gauge the semantic quality and relevance of the chatbot responses. Another important direction of research is the exploration of causal AI techniques to improve the quality of the responses it generates. Causal AI is an area of machine learning that builds models based on causal relationships rather than mere correlations.

This approach could be particularly useful in legal contexts, such as personal data protection, where understanding the cause-and-effect relationships between different elements of the law is crucial. One potential avenue to explore is the use of causal inference techniques to understand which elements of the training data have the most significant impact on the chatbot's performance.

By identifying these causal relationships, we could optimize the training process and focus on the most influential data elements. Further, integrating counterfactual reasoning within the chatbot may prove beneficial. Counterfactual reasoning is a core component of causal AI, enabling the model to consider alternate scenarios and outcomes, an ability particularly relevant in a legal context. For instance, understanding how a different data protection regulation could affect a certain scenario could be invaluable for users.

Acknowledgments. We would like to acknowledge the assistance provided by ChatGPT, developed by OpenAI. ChatGPT was used to enhance the syntax and wording of this paper, particularly in the editing and refinement of the manuscript. While ChatGPT offered suggestions and improvements, the responsibility for the scientific content and ideas presented in this paper remains solely with the authors.

References

1. Chin-Yew, L.: ROUGE: A package for automatic evaluation of summaries. In: Text Summarization Branches Out. Association for Computational Linguistics, pp. 74–81 (2004)
2. Denkowski, M., Lavie, A.: Meteor universal: Language specific translation evaluation for any target language. In: Proceedings of the Ninth Workshop on Statistical Machine Translation, pp. 376–380 (2014)
3. Gilardi, F., Alizadeh, M., Kubli, M.: ChatGPT outperforms crowd workers for text-annotation tasks. In: Proceedings of the National Academy of Sciences, vol. 120 (2023) doi: 10.1073/pnas.2305016120

4. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 328–339 (2018) doi: 10.48550/arXiv.1801.06146
5. Liu, B., Lin, T., Li, M.: Enhancing aspect-category sentiment analysis via syntactic data augmentation and knowledge enhancement. Knowledge-Based Systems, vol. 264, pp. 110339 (2023) doi: 10.1016/j.knosys.2023.110339
6. Loedeman, J., Stol, M. C., Han, T., Asano, Y. M.: Prompt generation networks for efficient adaptation of frozen vision transformers. In: Proceedings of the International Conference on Learning Representations (2022) doi: 10.48550/arXiv.2210.06466
7. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: Proceedings of the 36th Conference on Neural Information Processing Systems. Advances in Neural Information Processing Systems, vol. 35, pp. 27730–27744 (2022)
8. Papineni, K., Roukos, S., Ward, T., Zhu, W. J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002) doi: 10.3115/1073083.1073135
9. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
10. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G.: LLaMA: Open and efficient foundation language models (2023) doi: 10.48550/arXiv.2302.13971
11. Wang, J., Wang, C., Luo, F., Tan, C., Qiu, M., Yang, F., Shi, Q., Huang, S., Gao, M.: Towards unified prompt tuning for few-shot text classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2022) doi: 10.48550/arXiv.2205.05313
12. Watson, E., Viana, T., Zhang, S.: Augmented behavioral annotation tools, with application to multimodal datasets and models: A systematic review. AI Multidisciplinary Digital Publishing Institute, vol. 4, no. 1, pp. 128–171 (2023) doi: 10.3390/ai4010007

ISSN 1870-4069

Bayesian Classifier Models for Forecasting COVID-19 Related Targets Using Epidemiological and Demographic Data

Pedro Romero-Martínez[1], Christopher R. Stephens[1, 2]

[1] Universidad Nacional Autónoma de México,
Centro de Ciencias de la Complejidad,
Mexico

[2] Universidad Nacional Autónoma de México,
Instituto de Ciencias Nucleares,
Mexico

stephens@nucleares.unam.mx, pedro.romero@c3.unam.mx

Abstract. This paper proposes using Bayesian classifiers for predicting in space and time COVID-19 related targets such as infections, hospitalizations, intubations and deaths. In order to achieve this, Bayesian classifiers were developed and applied across a spatial grid, with each cell representing a municipality in Mexico. These models utilized open access epidemiological data between 2020 and 2021 published by the Mexican government's epidemiology agency and sociodemographic data from the 2020 national census of Mexico. Specifically, COVID-19 related targets are derived from epidemiological data and predictive features used in the model are extracted from socio-demographic and socio-economic data. Continuous variables from both datasets were discretized and represented as a finite set of presence-absence variables. These Bayesian models assign a "correlation" measure, known as score, to each variable with respect to the COVID-19 target. This implies that, we are able to identify profiles of the municipalities that are conductive to having COVID-19 related targets. The models generate two types of outcomes: (1) Spatiotemporal predictions of the abundance of COVID-19 targets are made using the Bayesian framework. (2) Predictions of number of individuals belonging to a given COVID-19 target for each municipality in a defined validation period. The utility of this framework is demonstrated by its strong performance in predicting the Mexican municipalities with the highest number of individuals in the top 10% of the target classes. Additionally, it provides reasonably accurate forecasts for the number of individuals within the target classes in each municipality.

Keywords: Epidemiology, SARS-Cov-2, COVID-19, Bayesian classifiers, Naive Bayes, complex adaptative systems, multifactoriality.

1 Introduction

The most recent pandemic was provoked by the SARS-Cov-2 virus. Since the first cases in December 2019 until November 2022, according to World Health Organization (WHO) [20], this disease has infected more than 634.5 million people and caused 6.5

millions deaths worldwide. The prevention and control of pandemics are of utmost importance from both the public health and scientific perspectives. Furthermore, the pandemic has demonstrated itself to be a Complex Adaptive System (CAS) as its evolution is contingent upon multiple factors which have changed and adapted over time as has the pathogen itself. One of the most important disciplines with which to study the pandemic is epidemiology "The systematic study of the distribution, causes and determinants (factors) of epidemiological states, risks or health-related events in specific populations, as in a geographical area, and its application to public health problems" [5].

The determinants play a crucial role in addressing the most relevant questions to understand about health phenomenon: when?, where?, why?, who?, what?, how?, etc. Therefore, epidemiology is a research discipline with an important public health component and a quantitative discipline encompassing descriptive and predictive perspectives. According to [14] "epidemiological intelligence is defined as the systematic compilation, analysis and communication of information aimed at detecting, verifying, evaluating and investigating events and risks for the public health, with the purpose of issuing an early alert".

In this context, it becomes crucial for decision makers to generate models about various aspects of the pandemic, interpreting the outcomes of these models in the real-world lead to lead to actionable insights. According to official Mexican government data, the COVID-19 pandemic has resulted in over 7 million infected people and more than 300 thousand deaths as of November 2022 in Mexico [6]. This pandemic has become the most extensively documented pandemic in world history, primarily owing advances in data collection, processing and storage capabilities achieved in recent years.

In Mexico, the Ministry of Health implemented a surveillance system for infections, which publishes daily the records obtained from a national network of hospitals. This database includes demographic data, comorbidities, clinical conditions and spatiotemporal attributes. Moreover, there are public datasets, such as the 2020 national census of Mexico, that can be included into the models as potential risk factors, processing them as presence-absence variables, as we will see later. In this work, Bayesian classifier models are generated to predict the number of individuals belonging to COVID-19 related targets, such as infected people and deaths.

The Bayesian models are computationally inexpensive, transparent, readily interpretable and have shown a good performance in a wide variety of problems [18, 19, 17], those are the main reasons to apply them. Unlike traditional SIRS-type epidemiological models, Bayesian classifier models enable the incorporation of a large number of variables, thereby capturing the high degree of multifactoriality of the pandemic.

2 Other Models

2.1 Differential Equations Models

In the 20th century, compartmental models were proposed for analyzing epidemics, consisting of an initial value problem, which involves ordinary differential equations (ODE) and initial conditions.

Although they are mathematically elemental, they help to develop the intuition for utilizing more sophisticated models. These SI(R)(S) models divide the population into groups, where the number of people in each group is time dependent: $S(t)$ is the number of **susceptibles**, $I(t)$ is the number of **infected** and $R(t)$ is the number of **recovered**. The equations contain some known parameters, such as the mortality rate μ, the contact rate λ and the recovery rate γ.

Some models have considered the number of births and deaths in the population by adding the term μN to the change in the susceptible group and subtracting a proportional amount from each group. In 1927, Kermack y McKendrick purposed the SIR model aimed at modelling specific epidemics, wherein individuals become immunized upon recovery:

$$\frac{dS}{dt} = -\lambda IS + \mu N - \mu S\,, \tag{1}$$

$$\frac{dI}{dt} = \quad \lambda IS - \gamma I - \mu I, \tag{2}$$

$$\frac{dR}{dt} = \quad \gamma I - \mu R, \tag{3}$$

where $S(0) = S_0 > 0$, $I(0) = I_0 > 0$, $R(0) = R_0 > 0$ and $S(t) + I(t) + R(t) = N$. However, there are certain diseases, such as COVID-19, in which individuals do not develop total immunity upon recovery. For such cases, we have the SIS model:

$$\frac{dS}{dt} = -\lambda IS + \gamma I + \mu N - \mu S, \tag{4}$$

$$\frac{dI}{dt} = \lambda IS - \gamma I - \mu I, \tag{5}$$

where $S(0) = S_0 > 0$, $I(0) = I_0 > 0$ and $S(t) + I(t) = N$. These types of models have been extensively studied, as seen in [13]. In the context of the COVID-19, numerous works have modeled the outbreak in different places, as evidenced in [2, 3, 4]. Furthermore, new versions of these models have been developed, by incorporating additional epidemiological states and transition rates between different groups [1].

Some other works identified certain deficiencies in the SI(R)(S) models, as seen in, [10]; in which, the authors utilized the SIR model to predict COVID-19 cases and deaths in Isfahan province of Iran, and discovered significant disparities between the long-term forecasts and the real cases and deaths.

Another common criticism of SI(R)(S) models is that they do not consider the multifactorial nature of a complex phenomenon such as an epidemic. For instance, these models do not incorporate factors beyond the simplified susceptible, infected etc. states, such as social, cultural, demographic, economic, ecological, geographical and others.

2.2 Machine Learning Models

Thanks to developments in computing and data storage capabilities in recent decades, applications of machine learning have proliferated across a variety of fields and disciplines.

There have been studies that utilized machine learning models to predict COVID-19 targets. The class of deep learning models learn patterns using neural networks with multiple neuron layers. A research group from Georgia Institute of Technology developed a deep learning model called DeepCOVID [12], aimed at making predictions about COVID-19 for each state in USA.

This deep learning framework utilized many data sources like COVID-19 epidemiological, COVID-19 tests, digital thermometer readings, mobility, social distancing measurements and viral load measurements. DeepCOVID was one of the first purely data driven and deep learning model and its results were very good in the short-term and trend performance. Another machine learning approach, utilized to interpret the COVID-19 cases and deaths over time as time series for a given place, is the attention mechanism models as applied to time series, weighting specific elements in the processing stage, as seen in [8].

In addition to the machine learning and SI(R)(S) models, some studies have presented hybrid models, combining the dynamics of compartmental models with machine learning techniques. For instance, in [3], interpretable encoders were utilized to incorporate covariates. Also in [16], a variation of SI(R)(S) is trained using weighted least squares. The main criticism for the deep learning and some of the hybrid models is their computational expense, which presents a challenge in generating real-time predictions, as running these models requires, special hardware as GPUs as well as their "black box" nature.

3 Bayesian Classifier Models

The general approach in this work is to employ a Bayesian framework, where the main objective is to estimate the conditional probability $P(C|\mathbf{X})$ for a given target class C, conditioned on a vector of attributes $\mathbf{X} = (X_1, X_2, \ldots, X_m)$. The general Bayesian approach possesses several advantages, as exemplified by Bayes' theorem :

$$P\left(C|\mathbf{X}\right) = \frac{P\left(\mathbf{X}|C\right)P\left(C\right)}{P\left(\mathbf{X}\right)}. \tag{6}$$

That relates the conditional probability $P(C|\mathbf{X})$, also known in this context as the posterior probability, with the likelihood function $P(\mathbf{X}|C)$, the evidence function $P(\mathbf{X})$ and the prior probability $P(C)$. $P(C|\mathbf{X})$ is referred as the posterior probability because it can be interpreted as a probability after the inclusion of the data associated with $\mathbf{X}$, providing a better estimation than the prior probability $P\left(C\right)$. Naturally, Bayes' theorem incorporates the phenomenon of adaptation, as the posterior probability can be re-calculated when new information $\mathbf{X}'$ become available, according to:

$$P\left(C|\mathbf{X}', \mathbf{X}\right) = \frac{P\left(\mathbf{X}'|\mathbf{X}, C\right)P\left(C|\mathbf{X}\right)}{P\left(\mathbf{X}'|\mathbf{X}\right)}. \tag{7}$$

Which determines how the previous posterior probability as a new prior is updated. Another advantage of employing the Bayesian approach is that it provides a natural framework for analyzing causality [11].

3.1 Naive Bayes

Given the impossibility in directly approximating $P(C|\mathbf{X})$ or $P(\mathbf{X}|C)$ in a frequentist sense it is necessary to find a method for estimating them. One well-known, tested and simple approximation is the called Naive Bayes method. It assumes that the variables $\mathbf{X} = (X_1, X_2, \ldots, X_m)$ are independent, thus:

$$P(\mathbf{X}|C) = \prod_{i=1}^{m} P(X_i|C), \tag{8}$$

$$P\left(\mathbf{X}|\overline{C}\right) = \prod_{i=1}^{m} P\left(X_i|\overline{C}\right), \tag{9}$$

where $\overline{C}$ the set complement of C i. Combining the equations (6) and (8) and the following approximation for the evidence function:

$$P(\mathbf{X}) = \prod_{i=1}^{m} P(X_i|C)\, P(C) + \prod_{i=1}^{m} P\left(X_i|\overline{C}\right) P\left(\overline{C}\right). \tag{10}$$

Then,

$$P(C|X) = \frac{\prod_{i=1}^{m} P(X_i|C)\, P(C)}{\prod_{i=1}^{m} P(X_i|C)\, P(C) + \prod_{i=1}^{m} P\left(X_i|\overline{C}\right) P\left(\overline{C}\right)}. \tag{11}$$

At this point, the score function $S(C, \mathbf{X})$ is introduced, which is a monotone function of $P(C|\mathbf{X})$ and can be interpreted as the odds ratio of C and its complement $\overline{C}$:

$$S(C, \mathbf{X}) = \ln\left(\frac{P(C|\mathbf{X})}{P\left(\overline{C}|\mathbf{X}\right)}\right) = \ln\left(\frac{P(C)}{P\left(\overline{C}\right)}\right) + \sum_{i=1}^{m} \ln\left(\frac{P(X_i|C)}{P\left(X_i|\overline{C}\right)}\right) = s_0 + \sum_{i=1}^{m} s_i(X). \tag{12}$$

Defining $s_0 := \ln\left(P(C)/P\left(\overline{C}\right)\right)$ and $s_i(X) := \ln\left(P(X_i|C)/P\left(X_i|\overline{C}\right)\right)$ for $1 \leq i \leq m$. The function $S(C, \mathbf{X})$ can be interpreted as a classifier, indicating that a record with profile $\mathbf{X}$ belongs to the target class C if $S(C, \mathbf{X}) > 0$ and it belongs to the class $\overline{C}$ if $S(C, \mathbf{X}) < 0$.

3.2 Generalized Naive Bayes

The Naive Bayes method is based on a strong assumption: the likelihood function can be completely decomposed, as shown in (8). Despite this supposition the Naive Bayes method has proven to be robust and surprisingly accurate, as demonstrated in [18]. However, this method can be generalized by employing an alternative factorization to (8), for considering correlations among the variables $\mathbf{X} = (X_1, X_2, \ldots, X_m)$. Let ξ be a partition of $\mathbf{X}$, that is, $\xi = \{\xi_1, ..., \xi_k\}$ where each ξ_j is a subset of $\mathbf{X}$ and they satisfy that $\{X_1, .., X_m\} = \cup_{j=1}^{k} \xi_j$ and $\xi_i \cap \xi_j \neq \emptyset$ for $i \neq j$.

Particularly, defining $\xi_j = \{X_j\}$ for $1 \leq j \leq m$, $\xi = \{\xi_1, ..., \xi_m\}$ represents the Naive Bayes approximation. Given a partition ξ the likelihood function factorization (8) can be generalized as:

$$P(\mathbf{X}|C) = \prod_{i=1}^{k} P(\xi_i|C), \quad \xi_i \in \xi. \tag{13}$$

Which, in general, differs from the Naive Bayes factorization. Analogous to (11) utilizing (13) instead of (8):

$$P(C|\mathbf{X}) = \frac{\prod_{i=1}^{k} P(\xi_i|C) P(C)}{\prod_{i=1}^{k_\xi} P(\xi_i|C) P(C) + \prod_{i=1}^{k_\eta} P\left(\eta_i|\overline{C}\right) P\left(\overline{C}\right)}, \tag{14}$$

where $\eta = \left\{\eta_1, ..., \eta_{k_\eta}\right\}$ is a partition different from ξ. Finally the score functions is generalized as:

$$S(C, \mathbf{X}) = \ln\left(\frac{P(C)}{P\left(\overline{C}\right)}\right) + \sum_{i=1}^{k_\xi} \ln\left(P(\xi_i|C)\right) - \sum_{i=1}^{k_\eta} \ln\left(P\left(\eta_i|\overline{C}\right)\right), \tag{15}$$

$$= s_0 + \sum_{i=1}^{k_\xi} S^C(\xi_i) - \sum_{i=1}^{k_\eta} S^{\overline{C}}(\eta_i), \tag{16}$$

where $S^C(\xi_i) := \ln\left(P(\xi_i|C)\right)$ and $S^{\overline{C}}(\eta_i) := \ln\left(P\left(\eta_i|\overline{C}\right)\right)$. Selecting $\eta = \xi$ in (14):

$$S(C, \mathbf{X}) = \ln\left(\frac{P(C)}{P\left(\overline{C}\right)}\right) + \sum_{i=1}^{k} \ln\left(\frac{P(\xi_i|C)}{P\left(\xi_i|\overline{C}\right)}\right). \tag{17}$$

This is a natural generalization of the Naive Bayes classifier.

4 Spatial Cells Ensemble

To calculate the score contributions we must have a statistical ensemble with which counts of N_C, N_{X_i} and N_{CX_i} can be made. We will consider two types of ensemble, starting with an ensemble of spatial cells - in the present case municipalities. Let R be a region in the two-dimensional plane, such as the surface delimited by Mexico in the map.

Suppose that $\mathcal{M} = \{c_i\}_{i=1}^{N}$ is a partition of R, that is, a set of subregions where $c_i \cap c_j = \emptyset$ for any $i \neq j$ and the union of these subregions is equal to R. $\mathcal{M}$ is defined as a mesh and the elements c_i are the cells. The set of municipalities in Mexico is a mesh for the region delimited by Mexico.

Then, a function $X_j : \mathcal{M} \rightarrow \{0, 1\}$ is called a presence-absence variable, we will say that X_j occurs in the cell c_i, if it satisfies that $X_j(c_i) = 1$. For a given mesh $\mathcal{M}$ and a set of presence-absence variables $\mathbf{X} = \{X_1, ..., X_m\}$, a target class is a subset C of $\mathcal{M}$. In this context, the Naive Bayes approximation (12) can be rewritten as:

$$S(C, \mathbf{X}) = \ln\left(\frac{N_C}{N - N_C}\right) + \sum_{i=1}^{m} \ln\left(\frac{N_{CX_i}/N_C}{(N_{X_i} - N_{CX_i}) / (N - N_C)}\right). \quad (18)$$

Because $P(C) = N_C/N$, $P(C) = (N - N_C)/N$, $P(X_i|C) = N_{CX_i}/N_C$ and $P\left(X_i|\overline{C}\right) = (N_{X_i} - N_{CX_i}) / (N - N_C)$, where N_C represents the number of cells belonging to the target class C and N_{CX_i} indicates the number of cells where both C and X_i co-occur. Clearly, if $N_C = 0$ or $N_{CX_i} = 0$ the score $S(C, \mathbf{X})$ is undefined, to avoid this possibility a standard Laplace term is applied [9]:

$$S(C, \mathbf{X}) = \ln\left(\frac{N_C}{N - N_C}\right) + \sum_{i=1}^{m} \ln\left(\frac{(N_{CX_i} + \alpha)/(N_C + 2\alpha)}{(N_{X_i} - N_{CX_i} + \alpha) / (N - N_C + 2\alpha)}\right). \quad (19)$$

There are several target classes related with COVID-19 that can be predicted utilizing the ensemble of cells. For example, the top 10% of cells with the highest number of COVID-19 cases during a training period. The Naive Bayes model assigns the score s_j to the variable X_j, and by using the expression (19) it is possible to calculate the score for each cell.

The score of each cell can be interpreted as a measure of correlation with the target class, cells with higher scores are more likely to belong to the target class. In the previous example, the cells with the higher scores during training period, are the more likely for belonging to the top 10% with the highest number of cases of COVID-19 in the subsequent period.

In order to capture the changes over time, three periods with the same length are considered: (1) the first period $t - 1$, (2) the training period t and (3) the validation period $t + 1$. For a given target class C, such as top 10% of cells with highest number of deaths, two special types of target classes $\hat{C}$ are defined as:

– **Improvement:** Cells that belong to C during $t - 1$ and do not belong to C during t.

– **Deterioration:** Cells that do not belong to C during $t - 1$ and belong to C during t.

By utilizing the target class $\hat{C}$ and presence-absence variables during the training period in the Naive Bayes method, it is possible to determine the improvement or deterioration of the target class for the validation period by identifying the cells with the highest scores.

5 Population Ensemble

In the population ensemble the fundamental element is not the cell, but the "person". Let N_i represents the population of the cell $c_i \in \mathcal{M}$. If $\mathcal{M}$ is the set of municipalities in Mexico, the N_i is the population of the municipality c_i. In this context, the target classes are defined based on the individuals, such as infected or death by COVID-19.

Table 1. Presence-absence variables derived from variable Female population.

Variable	Bin	Range
Female population	1	43.2%: 49.3%
Female population	2	49.3%: 50.0%
Female population	3	50.0%: 50.5%
Female population	4	50.5%: 50.9%
Female population	5	50.9%: 51.2%
Female population	6	51.2%: 51.5%
Female population	7	51.5%: 51.8%
Female population	8	51.8%: 52.2%
Female population	9	52.2%: 52.9%
Female population	10	52.9%: 60.0%

In this case, the population ensemble size coincides with the total population $N = \sum N_i$ and the presence-absence variables are based on the combined populations of the cells. The population ensemble enable us to predict the number of individuals in the target class by assigning a score to each individual using the expression (19), where N_C represents the number of people belonging to the target class C and N_{CX_i} indicates the number of people belonging to C and possessing the attribute X_i.

The higher the score of an individual, the more likely it is the individual belongs to the target class. Although for reasons of privacy it is not possible to create models which have socio-demographic and socio-economic variables documented for each individual over the whole population of Mexico, there are documented and publicly available variables defined over the set of municipalities of Mexico. In order to extend the use of the cells-defined (municipalities-defined) variables X_j to the entire population, we define the function $\hat{X}_j$ such that $\hat{X}_j = 1$ for individuals that are part of the population of any cell c_i that satisfies $X_j\,(c_i) = 1$.

For simplicity, the variables $\hat{X}_j$ will be just denoted by X_i. Using variables defined over the cells to make predictions, we assign the same score for a given variable to every individual within the same cell, as each individual within a given cell inherits the attributes of that cell. In order to determine the probability for each individual population ensemble, the score calculated for individuals is considered. Ranking the population based on their individual score and dividing into equally sized d sub-lists I_k, the probability for each sub-list is calculated as follows:

$$p_{I_k} = \frac{\text{number of individuals belonging to the target class } C \text{ within } I_k}{\text{number of individuals within } I_k}. \tag{20}$$

Just like in the cells ensemble the score depends on the period. Let's consider the scores and probabilities for each cell during the first and training period as $\left(S_i^{t-1}, p_i^{t-1}\right)$ and (S_i^t, p_i^t), the probability for each individual in the cell c_i computed in two ways:

Table 2. Predictions for the municipalities with the highest scores resulting from the model targeting deaths by COVID-19 in the population between 30 and 39. The first, training and validation periods are November 2020, December 2020 and January 2021, respectively.

State	**Municipality**	N_i	$\#C_i^t$	S_i^t	**pred.** $\#C_i^{t+1}$	$\#C_i^{t+1}$
Ciudad de México	Gustavo A. Madero	171225	21	55.483	52.26	36
Ciudad de México	Iztapalapa	281800	31	52.954	84.65	57
Ciudad de México	Tlalpan	107280	13	51.268	40.54	14
Ciudad de México	Iztacalco	61842	14	50.906	33.56	15
México	Cuautitlán Izcalli	84377	8	50.848	39.24	17

– Additive prediction: Let f be a regression model for the data (S_i^t, p_i^t), then define $\Delta p_i^t := f\left(S_i^t - S_i^{t-1}\right)$. The probability for each cell c_i in the validation period is given by $p_i^{t+1} := p_i^t + \Delta p_i^t$.

– Multiplicative prediction: $p_i^{t+1} := \dfrac{\#C_i^t}{\#C_i^{t-1}} p_i^t$.

Here, $\#C_i^t$ represents the number of the individuals in the target class within the cell c_i during the period t. For both types of predictions $\#C_i^{t+1} = p_i^{t+1} N_i$.

6 Model Validation

6.1 Spatial Validation

Given a training period t and a cells ensemble, the ensemble is randomly divided into two subsets: the training and the validation sets. The Bayesian model is trained using the training set, computing a score s_j for the presence-absence variables X_j during the training period. The score for each cell in the validation set is calculated using the variable scores s_j.

It is possible that certain cells may not have any calculated score variables associated with them, such cells are called nulls. The spatial validation aims to measure the model's ability to identify the validation cells in the target class. This purpose is analyzed using the recall defined as, $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ in each sub-list I_k, where TP is the number of true positives in the sub-list I_k, FN is the number of false negatives and the sub-lists are equally sized defined by ranking the validation cells by score.

6.2 Temporal Validation

Let t and $t+1$ be training and validation periods, respectively. The objective of the temporal validation in the cells ensemble is to measure the performance of the predictions over time. Similar to the spatial validation, the recall is analyzed for each sub-list I_k obtained by ranking the entire mesh by score and comparing it with the real data in the validation period. In this type of validation, the TP are cells in the target class during the validation period and belonging to I_k and the FN are the false negatives.

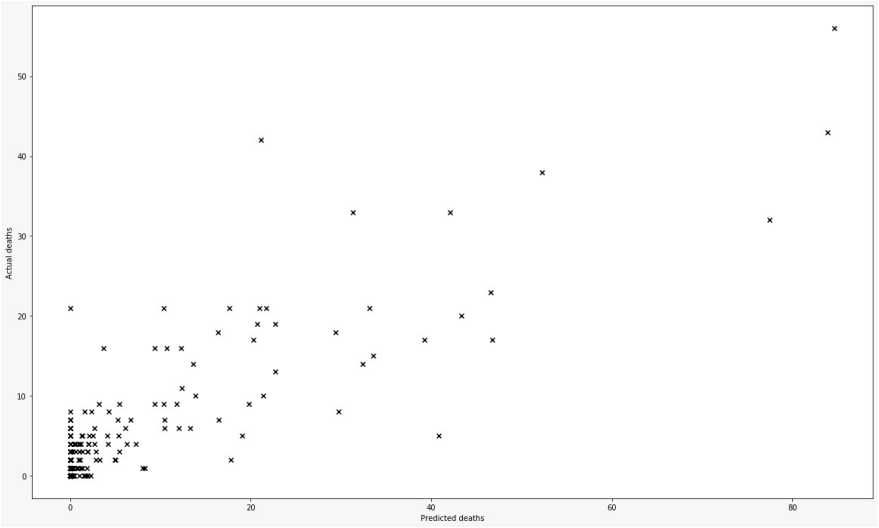


Fig. 1. Scatter plot showing the predicted values C_i^{t+1} versus the observed values of C_i^{t+1} for the predictions of the model in Table 2. The R^2 value is 0.8611.

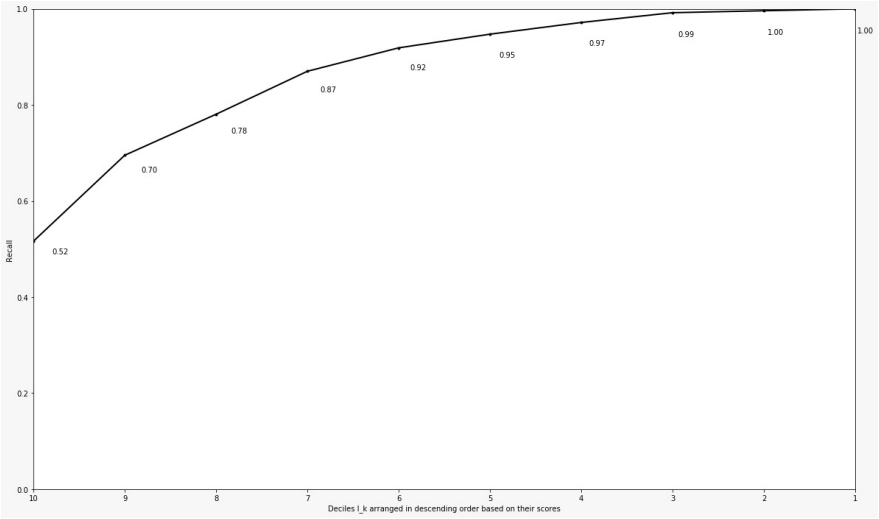


Fig. 2. Recall curve for the predictions of the model configuration in Table 2.

7 Data Processing

The data necessary to train the Bayesian models includes the target classes C for the specified periods, presence-absence variables X_j and the mesh $\mathcal{M}$ over the region R. This work focuses on Mexico as the region between the years 2020 and 2021 and the set of municipalities in Mexico as the mesh.

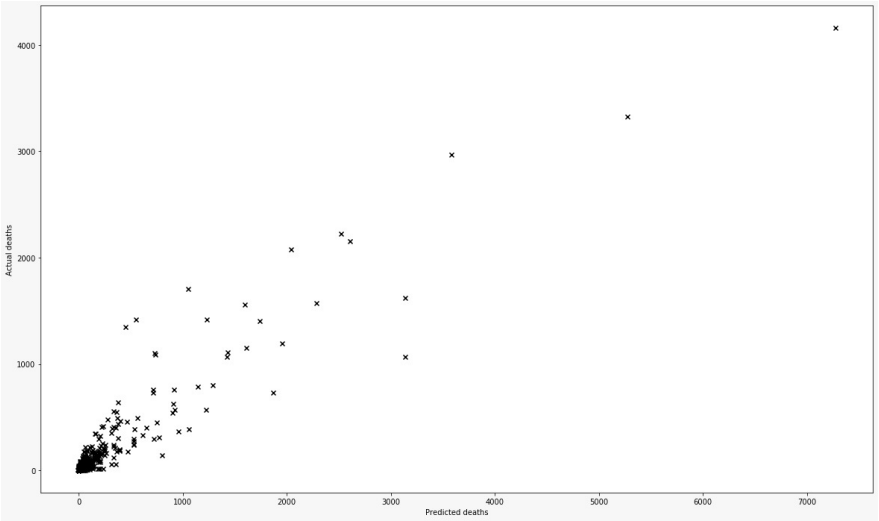


Fig. 3. Scatter plot of the prediction C_i^{t+1} versus the observed value of C_i^{t+1} for the predictions in Table 3, with an R^2 value of 0.9393.

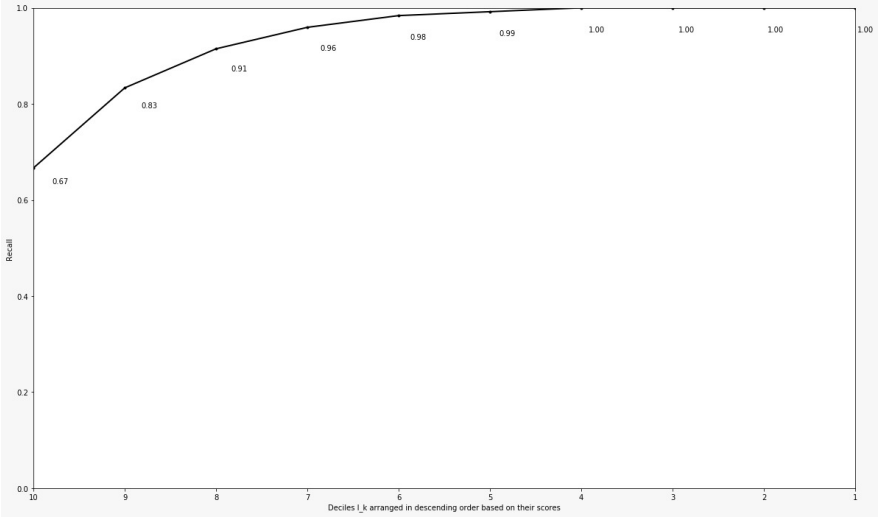


Fig. 4. Recall curve for the predictions of the model configuration in Table 3.

The presence-absence variables are derived from the processed variables of the 2020 national census of Mexico, while the target classes pertain to the epidemiological states of infection and death caused by COVID-19. The epidemiological states are obtained from the open COVID-19 database of the epidemiology agency of the Mexican government. This database is generated by the COVID-19 surveillance system, which publishes daily records reported by the hospital network in the country.

Table 3. Predictions for the municipalities with the highest scores resulting from the model targeting infections by COVID-19 in the population aged 60 years and older. The first, training and validation periods are November 2020, December 2020 and January 2021, respectively.

State	Municipality	N_i	$\#C_i^t$	S_i^t	**pred.** $\#C_i^{t+1}$	$\#C_i^{t+1}$
Ciudad de México	Álvaro Obregón	122319	2526	83.976	3630.63	2957
Ciudad de México	Gustavo A. Madero	203469	2488	83.107	5365.26	3416
Ciudad de México	Tlalpan	108894	1724	81.535	2557.30	2218
Ciudad de México	Venustiano Carranza	78964	1135	79.815	1998.50	1153
Ciudad de México	Coyoacán	126592	1416	79.397	3199.82	1615

In addition to capturing whether an individual is infected or not, it includes demographic profiles, comorbidity data, other clinical conditions, and spatial-temporal information at the daily and municipal level. For a given training period and target class, the open COVID-19 database provides the municipality information for each record that belongs to the target class. The open database where this data was obtained can be found at [15]. The presence-absence variables are derived from the 2020 national census database of the Mexican government [7].

The census database contains 180 variables with population and housing characteristics for different geographical levels. In particular, this study utilizes data at the municipal level. All census variables are integer-valued variables defined over the mesh of municipalities, and they are processed to generate presence-absence variables. First of all, as the variables are defined across the set of municipalities, and given the substantial diversity among municipalities, the variable values were normalized by dividing them by the population of each municipality.

Let $\mathcal{X}$ be a variable and d an integer value greater than 0. It is possible to obtain d presence-absence variables from the variable $\mathcal{X}$ as follows. Since the variable $\mathcal{X}$ is defined over $\mathcal{M}$, the rank is finite. Therefore, by sorting the rank, it is possible to divide it into d equally sized sub-ranks $(r_{j-1}, r_j]$. Each sub-rank defines a presence-absence variable $\mathcal{X}_j$ as follows: for every $c_i \in \mathcal{M}$, $\mathcal{X}_j(c_i) = 1$ if $r_{j-1} < \mathcal{X}_j(c_i) \leq r_j$.

This data processing transforms every variable into d presence-absence variables. Thus, fixing $d = 10$, 1800 presence-absence variables can be derived from census database. For example, the variable Female population is one of the 180 census variables defined across the set of municipalities, its the minimum value is 40 and the maximum is 953,783. For this specific variable, using the process described above, were derived 10 presence-absence variables presented in the Table 1.

8 Results

Several models have been generated for different configurations. In the population ensemble, the target classes considered were infection or death by COVID-19 for different age groups: 60 years and older, 50-59 years, 40-49 years, 30-39 years, and 18-29 years. Furthermore, each model had consecutive first, training and validation periods, each lasting 30 days. The target classes were defined based on two criteria: the COVID-19-related target and the age group.

For example, one target class included individuals aged 60 years and older who were infected by COVID-19, while another class included people aged 18 to 29 years who died from COVID-19. The variables utilized for the model training were derived from the 2020 national census database, as mentioned in the previous section, and all models had the same static presence-absence variables. The people ensemble models predicted the number of people in the target class for each municipality during validation period. Below, we present partial outcomes of two model configurations.

The first configuration, was considered people between 30 and 39 years old who died by COVID-19 as target class, using December 2020 as training period, the Table 2 displays the predicted and actual numbers of people in the target class (pred. $\#C_i^{t+1}$ and $\#C_i^{t+1}$ respectively) during the validation period for the municipalities in Mexico with the highest scores calculated in the model. Similarly, Table 3 shows the predicted and actual $\#C_i^{t+1}$ for the second example, where the target class consists of people aged 60 years and older who were infected by COVID-19, also using December 2020 as the training period.

The Figures 1 and 3 depict scatter plots generated using the predicted $\#C_i^{t+1}$ and the actual $\#C_i^{t+1}$ values for both model configurations. In both examples, the coefficient of determination R^2 is a high (near to 0.9), indicating that the 2020 census presence-absence variables effectively explain the number of people in the target classes using this methodology, and the predictions are reasonably accurate.

This framework assigns a score to each municipality as expressed by equation (19). Figures 2 and 4 demonstrate that this score is effective in predicting the municipalities that will belong to the top 10% with the highest number of individuals within the target class during the validation period, referred to as C_{10}^{t+1} for brevity. To achieve this, the entire list of municipalities is divided into 10 equally-sized sub-lists: $I_{10}, I_9, ..., I_1$, where I_{10} represents the top 10% of municipalities with the highest scores, and I_1 represents the bottom 10% with the lowest scores.

Figures 2 and 4 show that more than 50% of municipalities in C_{10}^{t+1} are included in I_{10}. Those municipalities within C_{10}^{t+1} and do not included in I_{10}, distributed across the remaining sub-lists I_k with $k \neq 10$, the Figures 2 and 4 display the growth percentage of municipalities in C_{10}^{t+1} and I_k with respect to I_{k+1}. In particular, in the second model configuration, as shown in Figure 4, it can be observed that 67% of the municipalities in C_{10}^{t+1} falls within I_{10} and all municipalities in C_{10}^{t+1} are accounted for in $I_{10}, I_9, ..., I_4$.

9 Conclusions and Discussion

While some of the developed models have incorporated variables from various domains (demographic, hospital infrastructure, mobility, social contact measures, etc.), they have been limited in quantity. Considering the complexity of the COVID-19 pandemic, which depends on numerous factors, it is important to include as many variables from relevant domains as possible to accurately model the reality.

Unlike the SI(R)(S) models, the Bayesian approach allows for the consideration of variables other than just the time series of infected and deceased individuals in making predictions.

In general, the reviewed literature agrees that the generated predictions are intended to support public health decision-makers in formulating more informed policies. However, very few models provide a measure of the factors most correlated with the target class of COVID-19 (infected, hospitalized, deceased, etc.), which would provide more specific guidance on the necessary actions to be taken.

In contrast to certain models, such as neural networks, which demand specialized hardware like Graphics Processing Units (GPUs) for real-time predictions due to intensive calculations during training, our proposed approach does not necessitate specific hardware and boasts reasonable training times.

The model has high practical utility for public health decision-makers, as indicated by its high R^2 value. This suggests that its predictions can provide valuable insights into what can be expected for the upcoming period. Ranking municipalities based on their scores offers a valuable means of identifying the municipalities that are likely to belong to the top 10% with the highest population within the target class during the validation period.

Acknowledgments. We sincerely thank PAPIIT, a research and technological innovation support program at UNAM. PAPIIT has generously supported numerous projects undertaken by the Chilam laboratory, and this paper is an outcome of our research efforts within the Chilam laboratory.

References

1. Acuña-Zegarra, M. A., Santana-Cibrian, M., Rodriguez-Hernandez-Vela, C. E., Mena, R. H., Velasco-Hernández, J. X.: A retrospective analysis of COVID-19 non-pharmaceutical interventions for Mexico and Peru: A modeling study. Cold Spring Harbor Laboratory Press, (2022) doi: 10.1101/2022.12.19.22283668
2. Anastassopoulou, C., Russo, L., Tsakris, A., Siettos, C.: Data-based analysis, modelling and forecasting of the COVID-19 outbreak. PLOS ONE, vol. 15, no. 3, pp. e0230405 (2020) doi: 10.1371/journal.pone.0230405
3. Arik, S., Li, C. L., Yoon, J., Sinha, R., Epshteyn, A., Le, L., Menon, V., Singh, S., Zhang, L., Nikoltchev, M., Sonthalia, Y., Nakhost, H., Kanal, E., Pfister, T.: Interpretable sequence learning for COVID-19 forecasting. In: Proceedings of the 34th Conference on Neural Information Processing Systems. Advances in Neural Information Processing Systems, pp. 1–12 (2021)
4. Chen, Y. C., Lu, P. E., Chang, C. S., Liu, T. H.: A time-dependent SIR model for COVID-19 with undetectable infected persons. IEEE Transactions on Network Science and Engineering, vol. 7, no. 4, pp. 3279–3294 (2020) doi: 10.1109/tnse.2020.3024723
5. Dicker, R., Coronado, F., Koo, D., Gibson-Parrish, R.: Principles of epidemiology in public health practice. U.S. Department of Health and Human Services (2012)
6. Gobierno de México: COVID-19 México (2022) datos.covid-19.conacyt.mx
7. INEGI: Censo de población y vivienda (2020)
8. Jin, X., Yu-Xiang, W., Yan, X.: Inter-series attention model for COVID-19 forecasting (2021)
9. Langou, J.: Translation and modern interpretation of laplace's théorie analytique des probabilités (2009)
10. Moein, S., Nickaeen, N., Roointan, A., Borhani, N., Heidary, Z., Javanmard, S. H., Ghaisari, J., Gheisari, Y.: Inefficiency of SIR models in forecasting COVID-19 epidemic: A case study of Isfahan. Scientific Reports, vol. 11, no. 1 (2021) doi: 10.1038/s41598-021-84055-6

11. Neuberg, L. G.: Causality: Models, reasoning, and inference. Econometric Theory, vol. 19, no. 4 (2003) doi: 10.1017/s0266466603004109
12. Rodríguez, A., Tabassum, A., Cui, J., Xie, J., Ho, J., Agarwal, P., Adhikari, B., Prakash, B. A.: DeepCOVID: An operational deep learning-driven framework for explainable real-time COVID-19 forecasting. vol. 35, pp. 15393–15400 (2021) doi: 10.1609/aaai.v35i17.17808
13. Satsuma, J., Willox, R., Ramani, A., Grammaticos, B., Carstea, A.: Extending the sir epidemic model. Physica A: Statistical Mechanics and its Applications, vol. 336, no. 3-4, pp. 369–375 (2004) doi: 10.1016/j.physa.2003.12.035
14. Secretaría de Salud: Manual de operación para las unidades de inteligencia epidemiológica y sanitaria (2021) epidemiologia.salud.gob.mx/gobmx/salud/documentos/manuales/39_Manual_UIES.pdf
15. Secretaría de Salud: Datos abiertos dirección general de epidemiología (2024)
16. Srivastava, A., Prasanna, V. K.: Learning to forecast and forecasting to learn from the COVID-19 pandemic (2020) doi: 10.48550/ARXIV.2004.11372
17. Stephens, C. R., González-Salazar, C., Romero-Martínez, P.: Does a respiratory virus have an ecological niche, and if so, can it be mapped? Yes and yes. Tropical Medicine and Infectious Disease, vol. 8, no. 3, pp. 178 (2023) doi: 10.3390/tropicalmed8030178
18. Stephens, C. R., Huerta, H. F., Linares, A. R.: When is the Naive Bayes approximation not so naive? Machine Learning, vol. 107, no. 2, pp. 397–441 (2017) doi: 10.1007/s10994-017-5658-0
19. Stephens, C. R., Sierra-Alcocer, R., González-Salazar, C., Barrios, J. M., Salazar-Carrillo, J. C., Robredo-Ezquivelzeta, E., del Callejo-Canal, E.: SPECIES: A platform for the exploration of ecological data. Ecology and Evolution, vol. 9, no. 4, pp. 1638–1653 (2019) doi: 10.1002/ece3.4800
20. World Health Organization: Coronavirus (COVID-19) dashboard (2022) covid19.who.int

ISSN 1870-4069

Predicting the Demand for Services at a Government Institute of Health in Mexico

Abraham Barroso, Noé Méndez,
Hiram Ponce

Universidad Panamericana,
Facultad de Ingeniería,
Mexico

{0264915, 0264134, hponce}@up.edu.mx

Abstract. Medical care is one of the issues that afflict the public health Mexican institutes' right holders on a daily basis, due to lack of personnel or waiting for care for long periods of time, so this paper seeks to support the government agency with the use of data and new technologies for better decision-making, through the use of machine learning and cloud computing technologies. For this reason, we have used linear regression models for our prediction tasks and compared their predictive power, in order for the institution to make a first approach and see the advantage of using new technologies and make more intensive use of them. Our results show that it is necessary to contemplate a greater number of data for more precise predictions, but it is something that the institution is not contemplated in the short time.

Keywords: Linear regression, medical services, helth institute, poor attention.

1 Introduction

Health system in Mexico suffers from limit capacity, lack of health professionals, lack of supplies and medicines, as well as some poorly implemented policies, which make trying to be attended a bad experience and in many cases discouraging due to these deficiencies.

Since the Mexican Government has different institutions that provide health services to the population, it is necessary to select one to start with, so the case study will focus on the Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado (ISSSTE), which is a parastatal entity of the Mexican Government that provides health and social security services to State workers and their families; providing services to 13.5 million beneficiaries in the country, which represents approximately 9% of the total population entitled to them in Mexico.

Public health institutes in Mexico have an important area of opportunity to make use of the large amounts of data that are generated every day in the different hospitals and services (from prescription refills, inventory management, hospital admissions and discharges, absenteeism, hiring, dismissals and retirements of medical and administrative personnel, etc.) in order to make better decisions and implement policies that lead to an improvement in the care and services provided to the population [4, 5].

Table 1. Datasets used in the experiment.

Description	Type
Federal Entity	string
Medical Unit Code	string
Name of the Medical Unit	string
Type of Medical Unit	string
Level of Care	string
Service/Specialty	string
Number of Consultations	integer
Service	string
Type of Consultation	string

The above through the use of technologies such as artificial intelligence and cloud computing services [1, 8], allowing to predict and anticipate the demand for these services to improve the care provided to beneficiaries, seek the correct allocation and recruitment of medical staff and implement measures and policies to mitigate the social discontent caused by poor or no care and have gained relevance in recent years.

Therefore, the interest and objective, as the first scope of this work, is to focus on the prediction of the demand for services and patient care, because it has become a controversial issue in the health institutions of the Mexican Government, especially in those that have to do with care specialties or involving specialized and complex procedures, with the need for the use of devices and high medical technology and the participation of a multidisciplinary team in some cases. This work aims to apply machine learning models for predicting the number of services required in the medical units of ISSSTE.

The aforementioned is sought to be achieved through the use of information obtained from the year 2022 from the institute, the use of machine learning models and an architecture based on cloud computing, which will allow us, at some point, to replicate it to other health institutions. The rest of the paper is as follows. Section 2 describes the dataset. Section 3 presents the proposal of the work. Section 4 includes the experimental design and Section 5 shows the results and discussion. Finally, Section 6 concludes the work.

2 Description of the Dataset

The dataset contains [3] information of number of consult for service an medical unit, as shown in Table 1 and Table 2. We have 35 federal entities (includes subdivisions by region for Mexico City), 10 types of medical units, 112 medical units and 3 levels of care. It is very important that we study the distribution of the response variable, since, at the end of the day, this is what we are interested in predicting. Its distribution is visualized in Fig. 1, and we will apply a logarithmic and square root transformation to see its distribution from different perspectives.

Table 2. Catalog of types of medical units.

ID	Description	Level	Number of Medical Units
CMCT	Medical Office in the Workplace	1	47
CAF	Family Care Office	1	391
UMF	Family Medicine Unit	1	402
CMF	Family Medicine Clinic	1	91
CMFEQ	Family Medicine Clinic with Specialty and Operating Room	1,2	16
CE	Specialty Clinic	1,2	6
CEQ	Specialty Clinic with Operating Room	2	5
CH	Hospital Clinic	2	72
HG	General Hospital	2	26
HR/HAE	Regional Hospital / High Specialty Hospital	2,3	14
CMN	National Medical Center	3	1

Table 3. Datasets statistics for variable "Consultas".

Variable	count	mean	Std	min	25%	50%	75%	max
Consultations	9408.00	1598.31	4719.47	1.00	182.00	551.00	1364.00	96506.00

With the above and with the support of Python we can evaluate which distribution fits our data, because some of the machine learning models need a specific distribution and in our case chi-square is the one that best fits our data. In Table 3, we show the statistical data for our numerical predictor variable. We generated a chart to show the distribution of the number of medical consultations by "Federal Entity", as depicted in Fig. 2. In the Table 4 we show the statistical data for our categorical variables.

3 Description of the Proposal

We adopt the general workflow of machine learning for tackling the problem, as summarized in Fig. 3. The details are described below.

Dataset Selection. As a first step, only the values were taken from the information provided by ISSSTE.

Cleaning and Adjustment. Subsequently, the headings were constructed as follows: We noticed that each specialty has 4 columns: First Time, Subsequent, Visit, Total. When we had this situation where what we needed was to distinguish each column in each specialty, a concatenation was made between that column and the specialty. For example, the column "First Time" appears in both Allergology and Anesthesiology, so they were as follows: Allergology_First_Time and Anesthesiology_First_Time.

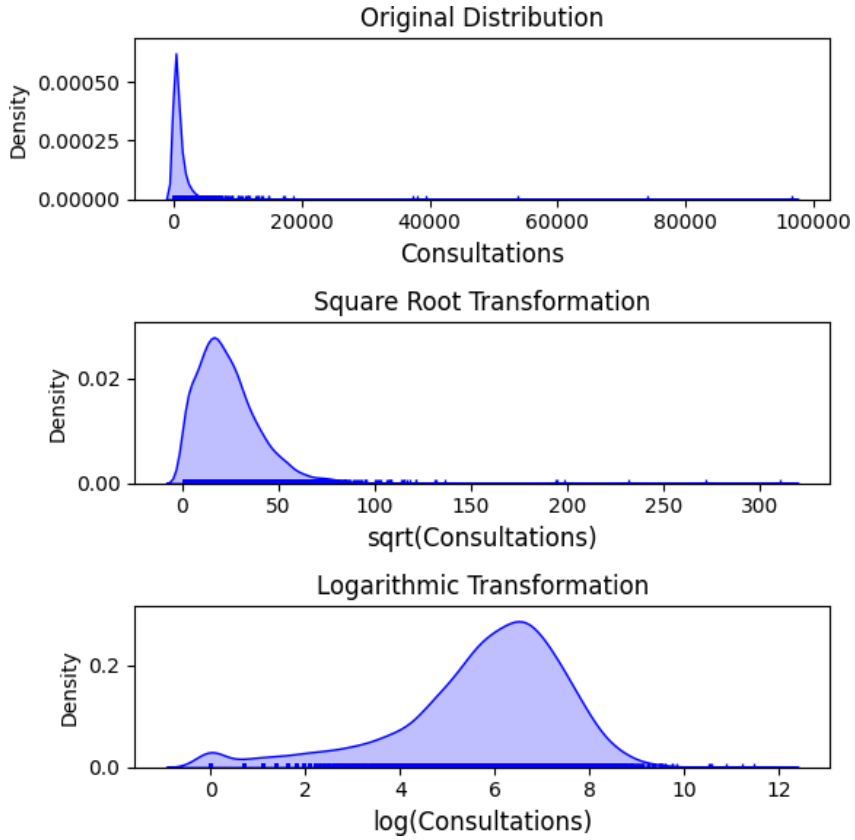


Fig. 1. Distributions.

Once this process was completed, a filter was implemented to eliminate the totals that were presented between the same information by Delegation. In this way, we obtained an optimal dataset to work with and begin to perform exploratory analysis.

Normalization Process. It is a method in which the values in a numeric column change so that the data set has a common scale, without distorting the differences in the ranges of values or losing information, and this may be a necessary activity for use in certain algorithms. For our data normalization we will use StandardScaler, a class that standardizes the data by removing the mean and scaling the data so that its variance is equal to 1.

Null Values. To handle null values we will use Sklearn's SimpleImputer function, which allows us to substitute null values for other values according to various strategies available in it.

Categorical Variables. For our work and to code our variables we will use OneHotScaler, whose strategy is based on creating a binary column (with values 0 or 1) for each single categorical value and places a 1 in the corresponding column where a value is present, leaving the rest of the columns with value 0.

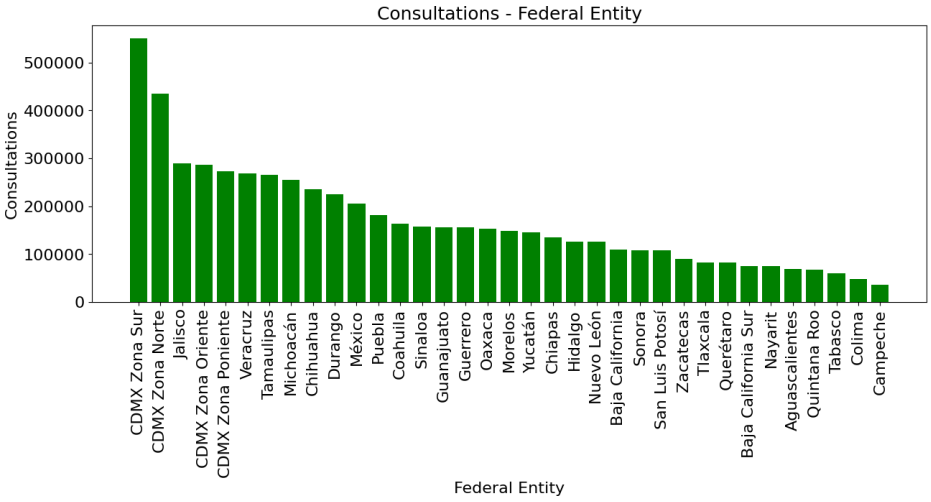


Fig. 2. Number of medical consultations.

Table 4. Datasets statistics for categorical variables.

Variable	Count	Unique	Top	Freq
Federal Entity	9408.00	35	Veracruz	21600.00
Key	9408.00	1071.00	001-204-00	270.00
Name	94080.00	1021.00	Tuxpan	1080.00
Type	9408.00	11.00	UMF	108540.00
Level	9408.00	3.00	1st	251370.00
Service	9408.00	90.00	Continuous.Admission.Adults	3213.00
Type of Consultation	9408.00	3.00	First.time	96390.00

Solution Arquitecture. As you can see in Fig. 4, the architecture includes components in the Microsoft Azure cloud, with the idea of a much more agile deployment, easy to scale and start with small scopes. Given that the information with which the predictions will be generated is non-sensitive, it is possible to take advantage of having an architecture in the cloud with security.

Models. We select four well-known machine learning models for the study:

– **Multiple linear regression [6]:** allows us to generate a linear model in which the value of our dependent variable (also known as response (Y)) and which is determined from a set of variables, known as independent or predictors (X1, X2, X3...). This is a variation or extension of simple linear regression.

In our case we will use it to predict the value of the dependent variable, but it is not the only thing that can be done with this model as it can allow us to see how the response variable is influenced by the independent variables.

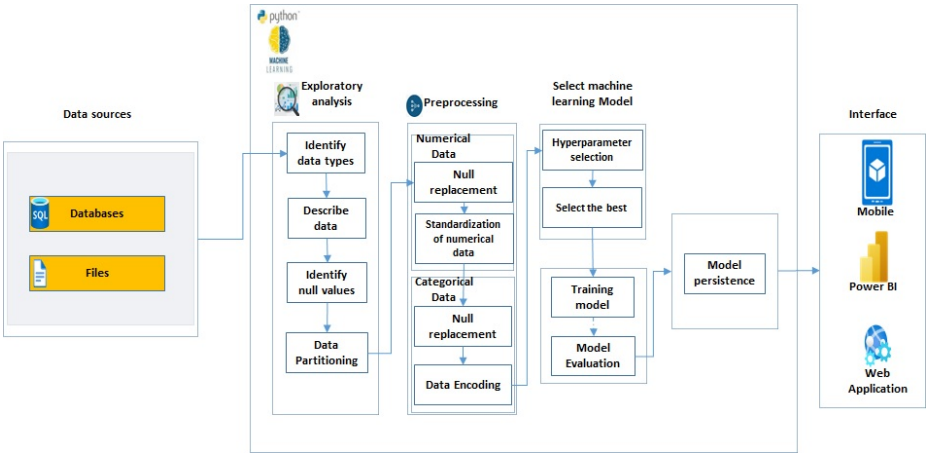


Fig. 3. General flow of our machine learning process.

The multiple linear model has the following equation (1):

$$Y_i = (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_n X_{ni}) + e_i. \tag{1}$$

– **Lasso regression [6]:** (Least Absolute Shrinkage and Selection Operator) is a linear model that penalizes the coefficient vector by adding its L1 norm to the cost function:

$$\text{Minimize } \theta : \sum_{i=1}^{N} [y_i - f(x_i, \theta)]^2 + \lambda \sum_{j=1}^{M} |\theta_j|. \tag{2}$$

This model has the characteristic of generating "sparse coefficients": which are vectors of coefficients in which most of them take the value zero. So the model considers ignoring some of the predictive features, which can be considered a type of automatic feature selection.

The model by performing feature exclusion seeks to generate a model that is simpler to interpret and exposes the most important features of our data set. If there is a correlation gap between the predictive features, the Lasso model will tend to choose one of them at random.

– **Ridge regression [2]:** Also known as contracted regression or Tikhonov regularization, aims to regularize the resulting model and imposes penalties on the size of the coefficients of the linear relationship between the predicted characteristics and the target variable. The coefficients that are calculated in the model seek to minimize the sum of the squares of the residuals by penalizing them by adding the square of the L2 norm of the vector formed by the coefficients:

$$\text{Minimize } \theta : \sum_{i=1}^{N} [y_i - f(x_i, \theta)]^2 + \lambda \sum_{j=1}^{M} \theta_j^2. \tag{3}$$

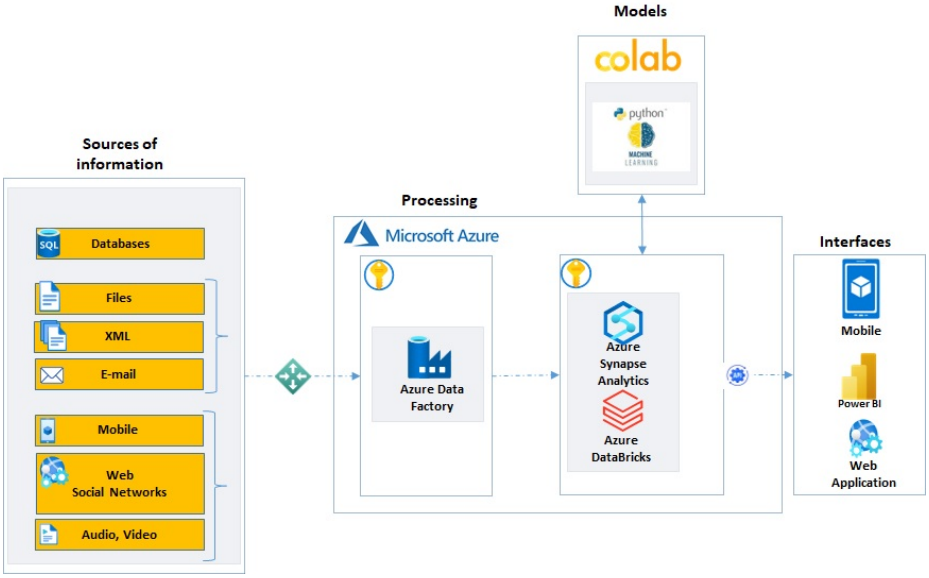


Fig. 4. Solution architecture.

In the formula of the model we have that λ is a parameter that controls the degree of penalization: the higher the value of λ , the lower the coefficients will be resulting more robust to collinearity.

– **Support Vector Regression (SVR) [7]:** this model is born from a variant of the Support Vector Machine (SVM) analysis model which is to perform classification tasks, however, the SVR model makes some minor changes in its definition.

For its use in regression cases, a tolerance margin (ϵ) is established near the vector and its purpose is to try to minimize the error, taking into account that part of that error is tolerated.

Equation for Linear SVR:

$$y = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)(x_i,\, x) + b. \tag{4}$$

Equation for Non-linear SVR:

$$y = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)(\varphi(x_i),\, \varphi(x)) + b, \tag{5}$$

$$y = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x_i,\, x) + b. \tag{6}$$

For hyper-parameter optimization we use grid search which performs an exhaustive search by evaluating all parameter combinations.

Table 5. Results of evaluation metrics.

ID	Model	Model	Model	SVR
	Linnear Regression	**Regression Lasso**	**Regression Ridge**	**SVR**
MSE	3.620 e+42	3693492.38	3563258.70	3907206.78
MAE	59.67 e+48	798.64	819.77	1976.66
$\mathbf{R}^2$	-9.53 e+37	0.02895	0.06319	-0.02723

This strategy has the disadvantage of requiring high consumption of computational resources when the number of data becomes too large, as well as the evaluation of regions that may be of little interest before evaluating more combinations, it uses cross-validation techniques for its operation.

4 Experimentation

For this work it was decided to use cross-validation, which consists of randomly dividing the observations obtained into k groups of equal size. One of the k groups is used as the validation set, while the remaining $k-1$ groups are used to train the model. The mean squared error (MSE) is calculated on the $k-1$ groups excluded from the model, this validation process is repeated k times because each group is used as a validator. So in the end we obtain k estimates of the mean squared error and calculate the overall estimate by averaging the k values of our linear regression using (7):

$$CV_k = \frac{1}{k}\sum_{i=1}^{k} \mathrm{MSE}_i \,. \tag{7}$$

For our experiments we use 80% of the data for training and 20% for validation of our models. As evaluation metrics, we consider the following ones (8)-(10):

Mean Absolute Error (MAE):

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} \mid y_i - x_i \mid}{n}. \tag{8}$$

Mean Squared Error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2. \tag{9}$$

R Squared Value (R^2):

$$R^2 = \frac{\mathrm{SSR}}{\mathrm{SST}} = 1 - \frac{\mathrm{SSR}}{\mathrm{SST}} = 1 - \frac{\sum_{i}^{n}(y_i - \hat{y_i})^2}{\sum_{i}^{n}(y_i - \bar{y_i})^2}. \tag{10}$$

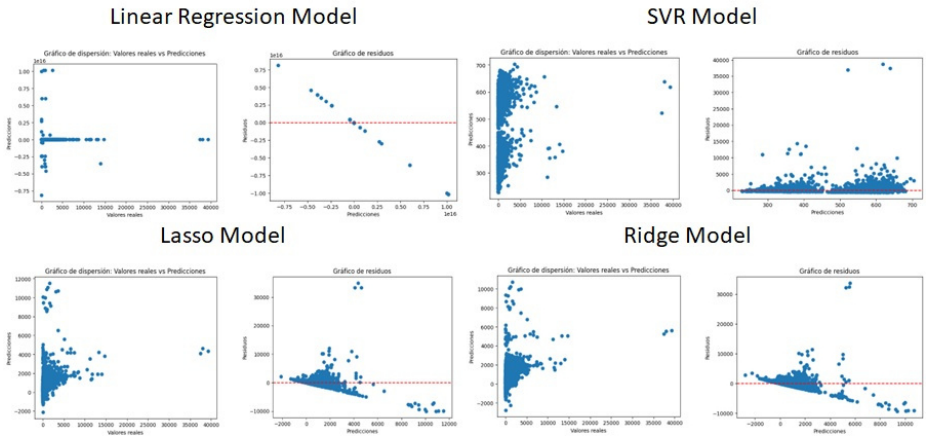


Fig. 5. Scatter plots and residual plots of the models used.

5 Results and Discussion

As part of our results we reviewed the coefficient of determination, also known as R^2 (R-squared), as a metric used to assess the quality of our regression models. This provides us with a measure of how well the model predictions fit the actual values of the target variable. The comparison of our models with respect to the value of the coefficient of determination ranges from 0 to 1, as shown below in Table 5.

As we can see our Lasso regression and Ridge regression models are close to 0 in our R^2 evaluation and indicate that the model is not able to explain the variability in the data and that the predictions are similar to simply using the mean value of the target variable. This suggests that the model is not adequate to represent the relationship between the predictor variables and the target variable.

In the case of SVR the R^2 value is negative, so our model performs very poorly in making predictions. Our model behaves in extremely poor ways and the errors are large compared to the variability of the data. Analyzing the Mean Squared Error (MSE) used to assess the quality of a regression model. Consider that the MSE is a measure of the variance or dispersion of the errors of our model and for our model the lowest MSE we have in the Ridge regression model is the one with the best fit to the data, since the errors are smaller and closer to zero.

On the other hand, the SVR and linear regression models have a higher MSE indicating that the models have a worse fit, this because the errors are larger and farther from zero. Finally we analyze the MAE which, as mentioned, is calculated by taking the absolute difference between the values predicted by the model and the actual values, and then calculating the average of these differences.

When interpreting the MAE for each of our models, we must take into consideration that this value is representing the average magnitude of the model errors under the same scale as our original data (number of queries) and as a difference from the MSE, the MAE does not square the errors, which allows us to keep the metric unaffected by outliers or large errors.

In our models, the Lasso Regression and Ridge model values have the lowest MAE metrics and indicate that the models have a better fit to the data, since the errors are smaller on average, while the linear regression and SVR model have high values that indicate that our models have a worse fit, due to the fact that the errors are larger on average. Lastly, Fig. 5 shows the scatter and residual plots of the models. We notice that the SVR model for the part of the residuals is behaving with homoscedasticity, while the rest of the models are showing heteroscedasticity.

6 Conclusions

This work aimed to study the performance of four machine learning models for predicting the number of services required in the medical units of ISSSTE. We adopted the general workflow of machine learning to approach our goal. By performing the evaluation of our models using the metrics proposed in our project (MAE,R^2, MSE), we have concluded that the results obtained by these currently have discrepancies to perform the prediction of some of the data and the metrics indicates a very low value in their prediction process. Given the above, it is necessary to:

– Evaluate and increase the number of data to improve the developed models as a first action, to subsequently evaluate the use of more features, provided that these do not introduce noise or that they are not significant and may affect their performance.

– To evaluate in a more exhaustive way the hyper-parameters that each of the models use and that were carried out in the development of the present project.

– Evaluate the use of other models, currently the machine learning area has a wide range of models (some with improvements of the base versions) that can support a better evaluation of data prediction and support better decision making.

When experimenting using grid search techniques for optimal hyper-parameters that maximize the performance of our models, we found that it can be a time-consuming and costly process in terms of computational processing, so as a continuation of this project, additional hyper-parameter search methods (e.g., Bayesian sampling) should be evaluated.

References

1. Cohen, T. A., Patel, V. L., Shortliffe, E. H.: Intelligent systems in medicine and health: The role of AI. Springer International Publishing (2022) doi: 10.1007/978-3-031-09108-7
2. Ehsanes Saleh, A. K. M., Arashi, M., Golam-Kibria, B. M.: Theory of ridge regression estimation with applications. John Wiley and Sons, Inc (2019) doi: 10.1002/9781118644478
3. Instituto de Seguridad y Servicios Sociales de los Trabajadores del Estado: Estadística anuarios (2022) www.issste.gob.mx/datosabiertos/anuarios/anuarios2022.html
4. King, Z., Farrington, J., Utley, M., Kung, E., Elkhodair, S., Harris, S., Sekula, R., Gillham, J., Li, K., Crowe, S.: Machine learning for real-time aggregated prediction of hospital admission for emergency patients. npj Digital Medicine, vol. 5, no. 1, pp. 104 (2022) doi: 10.1038/s41746-022-00649-y

5. Liu, Y., Qin, S.: An interpretable machine learning approach for predicting hospital length of stay and readmission. In: Proceedings of the 17th International Conference on Advanced Data Mining and Applications, vol. 13087, pp. 73–85 (2022) doi: 10.1007/978-3-030-95405-5_6
6. Montgomery, D. C., Peck, E. A., Vining, G. G.: Introduction to linear regression analysis. Wiley (2021)
7. Ozer, M. E., Sarica, P. O., Arga, K. Y.: New machine learning applications to accelerate personalized medicine in breast cancer: Rise of the support vector machines. OMICS: A Journal of Integrative Biology, vol. 24, no. 5, pp. 241–246 (2020) doi: 10.1089/omi.2020.0001
8. van-Houten, H.: El poder de la predicción: Cómo la IA puede ayudar a los hospitales a prever y gestionar el flujo de pacientes. Philips: Centro de noticias España (2021) www.philips.es/a-w/about/news/archive/standard/news/blogs/2021/20210906-the-power-of-prediction-how-ai-can-help-hospitals-forecast-and-manage-patient-flow.html

ISSN 1870-4069

A Comprehensive Review of Sign Language Translation Technologies Using Linguistic Approaches

Obdulia Pichardo-Lagunas, Bella Martinez-Seis,
Carlos Gómez-García

Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en
Ingeniería y Tecnologías Avanzadas,
Mexico

{opichardola, bcmartinez}@ipn.mx,
cgomez.egarcia@gmail.com

Abstract. In a society that seeks to be inclusive, communication between the deaf and hearing communities should be a priority, even so, knowledge of sign language between speakers is scarce, so the development of tools that simplify this communication is essential. It is important the development of software applications that allow the translation of Sign Language into a spoken language or in reverse. Most of the approaches display a sign for each word, the result is a signed sentence that significantly differs from the real signed language. Sign languages have their own grammar structure, which led us to analyze them with their own language components, which should be considered in Machine Translation. This paper describes studies that consider the syntactical component in machine sign language translation. We use a common procedure in the description of works in this field, including documents classified into two categories: Rule-based and Corpus-based. The works based on corpus are divided into Statistics and Hybrid Machine Translation, and Neural Machine Translation. It is important to use new technologies such as deep learning and neural networks in sign language translation systems. In addition to considering the different levels of linguistic analysis in translation.

Keywords: Automatic translation, sign language, syntactic translation.

1 Introduction

Deaf individuals use sign languages as their primary means of communication in daily life. There are more than 200 different sign languages in the world [15]. Communication with hearing individuals encounter barriers, primarily due to limited knowledge about sign languages among the hearing community. The study of sign languages has demonstrated that they are complex linguistic systems that allow people to communicate using their hands and vision to establish communication. A translation process is required to convert the spoken language of the hearing person into sign language for the deaf person, and vice versa (sign language to speech conversion). Automatic translation has emerged as a solution to overcome this language barrier

by automating the translation process. There are different approaches to address the issue of automatic translation between different sign languages and spoken languages, such as translating sign language to spoken language and vice versa, representing sign languages in written form using videos or avatars, and translation based on rules, statistics, and Machine Learning techniques, which have become widespread in recent years. However, in this document, we will focus only on research works that deal with automatic translation considering linguistic aspects rather than translating word-for-word.

1.1 Motivation

There are many factors that motivate this work, however we must recognize the need to develop computational tools that facilitate communication between the hearing and the silent community. Given the above, the need to disseminate existing work is essential to encourage researchers to develop new technologies and analyzes in this field. There are many isolated works regarding the field of automatic translation for sign language, however, there are also many approaches from which it has been addressed; this work seeks to compile these approaches. Finally, the review of these works seeks to document the final transition of the word-for-word translation carried out for years.

2 Fundamentals of Automatic Sign Language Translation

In this section we will address different concepts associated with the automatic processing of Sign Language.

2.1 Sign Language

Sign Language is a naturally occurring language that developed as results of the need to communicate among the Deaf communities. Sign language is a language that occurs in the visual-gestural modality, this means that it relies mostly on the use of hands, face, and upper torso. Like many other languages, Sign Language has undergone many transformations throughout its history; this essays traces and details the history or the development of sign language [36].

2.2 Machine Translation

Machine translation (MT) involves to translate a text from one language to another without human intervention. Instead of simply translating the text literally, modern machine translation aims to communicate the complete meaning of the original text in the target language. To achieve this, it analyzes all elements of the text and recognizes how words relate to each other. There are different approaches to machine translation that could be grouped into: Rule-based Machine Translation and Corpus-based Machine Translation. The rule-based machine translation could be direct-based (word by word), interlingua-based (independent interlingua representation), or transfer-based (dependent interlingua representation).

On the other side, corpus-based machine translation includes statistical, example-based, hybrid, and neural. Some authors [22] consider Neural Machine Translation as a third main type of machine learning; because of the data needed for training we consider it as a class inside corpus-base machine translation.

Rule-based Machine Translation. This type of translation requires language specialists to develop linguistic rules and dictionaries for specific topics or domains. Rule-based machine translation utilizes these resources to accurately translate specialized content. The process consists of the following steps: first the machine translation software analyzes the input text and creates an intermediate representation; second, using the grammatical rules and dictionaries as references, the software converts the intermediate representation into the target language.

Corpus-based Machine Translation. Corpus-based approach n (also referred as data driven machine translation) automatically extracts the knowledge by analysing translation examples from a parallel corpus built by human experts. The corpus-based approach is classified into the following sub-approaches:

– **Statistical Machine Translation.** Unlike rule-based translation, this type of translation uses Machine Learning (ML) techniques to translate texts. ML algorithms examine large amounts of previous human translations in search of statistical patterns. Then, when faced with a new source text, the software makes an intelligent guess on how to translate it. This is achieved by making predictions based on the statistical probability of a specific word or phrase appearing next to another word or phrase in the target language.

– **Hybrid Machine Translation.** Hybrid machine translation tools employ multiple machine translation models within a single software system. The hybrid approach is utilized to enhance the performance of a single translation model. This method typically integrates rule-based and statistical machine translation subsystems. The ultimate translation output is a combination of the outputs from all subsystems [3].

– **Neural Machine Translation.** Neural machine translation harnesses the power of artificial intelligence to acquire language knowledge and enhance it iteratively through a specific machine learning technique known as neural networks. It frequently collaborates with statistical translation methods to achieve its objectives.

2.3 Machine Translation for Sign Language

According to Yin [33] the translation of Sign Languages comprises at least the following tasks: detection, identification, segmentation, recognition, translation, and production. The most advanced studies on Sign Language Translation include the detection task that refers to identifying which sign language is being used. However, we must consider that most of the work carried out in this sense is carried out in isolation for specific Sign Languages. Sign-by-sign translation marked the first steps of automatic sign language translation. The most common was to assign a previously marked label to each sign. This label, which we call a gloss, is a specific transcription of each sign in sign language. The gloss is a notation mechanism to facilitate the representation of signs for study.

2.4 Analysis of Linguistic Levels

Linguistics, as a field of study, encompasses five main branches: phonology, morphology, syntax, semantics, and pragmatics. These branches represent distinct areas of language analysis, each focusing on specific aspects of communication.

- **Phonology** refers to the study of the sounds of a language. Every language has a set of sounds and logical rules for combining those sounds to create words. The phonology of a language refers to sounds and the processes used to combine them in spoken language.
- **Morphology** is the study of the internal structure of the words of a language including suffixes, prefixes, or infixes to create new words. The morphology of a language refers to the word-building rules speakers use to create words.
- **Syntactic** is the study of sentence structure. Any language has its own rules for combining words to create sentences. The syntactic analysis describes the rules that speakers use to put words together to create meaningful phrases and sentences.
- **Semantics** is the study of meaning in language. Linguists attempt to identify how the speakers of a language discern the meanings of words in their language and the logical rules speakers apply to determine the meaning of phrases, sentences, and paragraphs. The meaning of a word can depend on the context in which it is used.
- **Pragmatics** is the study of the social use of language. A linguistic analysis of pragmatics can describe the social aspects of the language sample being analyzed.

3 Methodology

The paper focuses on automatic translation at the syntactic level of sign languages. Discarding from the literature those that remain in the morphological component, since it refers to a word-for-word translation ignoring the syntactic structure of the languages. It should be noted that no automatic translation works in sign language were found considering the semantic and pragmatic components since the approach described in Section 5.1.

The phonological component is discarded because it is related to the execution of the sign, that is, the gloss; At this level, the signs typically captured from the video are recognized; at the other end of the translation, the gloss allows the generation or reproduction of the sign through avatars or images. It is important to denote the deep work in this area. There is a recent increase in research in this regard, including mainly neural networks.

Researchers use traditional Neural Networks like feed-forward back propagation network [34], but also new approaches like CNN for the sign recognition like [8] with two CNN, [1] uses CNN and LSTM, and others [31, 27] use RNN like LSTM and GRU. For the sign generation Advesrsalial Neural Networks [51] and GAN [54, 35] have emerged.

Table 1. Corpora comparison.

Corpus Name / Work Title	No. Sentences	No. Words	Languages Involved
RWTH-PHOENIX - Weather	1980 in DGS, 1489 in German	911 in DGS, 1489 in German	German Sign Language German
ISLTranslate	31k	11k	Indian Sign Language-English
ASLG-PC12	Over one hundred millions of pairs sentences	-	American Sign Language-English
Multimedia Corpora of Mexican Sign Language (MSL) with Syntactic Function	-	1505 words in Spanish related to 1019 videos of signs	Spanish-Mexican Sign Language
Translating Speech to Indian Sign Language Using Natural Language Processing	A video DB created by the authors. The DB contains 1000+ videos and open-source ISL videos	-	Indian Sign Language - English
Linguistic Restrictions in Automatic Translation from Written Spanish to Mexican Sign Language	-	206 signs with synonyms and 1790 signs from Manos con voz Mexican Sign Language dictionary	Spanish-Mexican Sign Language.
KArSL	-	502 signs that cover 11 chapters of ArSL	Arabic Sign Language
LSE-Sign	-	2,400 individual signs taken from standardized LSE dictionary	Spanish Sign Language
ISL-CSLTR	100 spoken language sentences	1036 word level images	Indian Sign language - Indian
ASL-LEX	-	nearly 1000 signs	American Sign Language

4 Sign Language Datasets

The creation of a spoken language to sign language translator faces significant challenges in obtaining sample translation examples. Limited interpreter availability, scarcity of sign language studies, and substantial grammatical differences between spoken and sign languages contribute to this difficulty.

Moreover, the lack of standardization poses a challenge, as different sign languages may have distinct grammatical rules. Another challenge arises from segmenting sentences in sign language, which requires expert sign language proficiency to accurately identify the start and end of signs. This often necessitates manual frame segmentation in datasets, particularly in videos with sign language interpreters, requiring the assistance of a sign language expert. The demanding nature of this task, along with the need for numerous examples, makes dataset collection labor-intensive and costly. The Table 1 shows a description of some of the corpora used for translating spoken language to sign language.

5 Natural Language Processing in Automatic Sign Language Translation

Natural Language Processing (NLP) is a discipline that studies language issues in human-to-human and human-to-machine communication [5]. It studies Automatic Translation, also known as Machine Translation, at the different language levels or components wich are describe in the next section. We focus this study on Syntactic and Semantic levels for Automatic Sign Language Translation.

5.1 Language Components and Translation in Sign Language

Language components are phonology, morphology, syntax, semantics, and pragmatics. We identify those components in sign language. We focus on Mexican Sign Language (MSL) for the examples.

Phonologic Component. Sign languages have no phonemes but we can do an analogy for the phonological component. Oral languages consist of a series of successive sound elements, while visual signs have a series of simultaneous constituents. It has [48]:

– Queiremas: Involves hands and finger positions, this is the one that most people identify.
– Toponemas: The 25 body zones where the sign is done. For example, the sign of pain usually points to the body part that hurts:
– Kinemas: 18 different movements and the number of times those are done. For example, the sign of person involves one movement from top to bottom but if the sign is plural (persons) the movement is done three times.
– Kineprosemas: 6 directions of the sign. For example, the sign of help has a different movement depending on who gives and who receives the help.
– Queirotropemas: There are 9 different palm orientations.
– Prosoponema: Involves facial expressions.

According to the tasks of the translation of Sign Languages proposed by [33], this phonological component has to be identified when doing a translation from signs to oral language; also, it is obtained for the production of the sign when doing a translation from oral to sign language.

Table 2. Classification of works by the type of automatic translation.

Type of Machine Translation	References
Rule-base Machine Translation.	[17, 58, 60, 49, 11]
	[4, 30, 46, 2, 45]
	[13, 12, 44, 37, 25]
	[29, 19, 26, 56]
Corpus-based: Statistical and Hybrid Machine Translation.	[59, 39, 53, 50]
Corpus-based: Neural Machine Translation.	[55, 8, 32, 54, 6]

Morphologic Component. A sign is the union of a concept and an acoustic image. A morphological analysis gives a direct translation where each word is represented by a sign, or each sign is represented by a word. Translation at this level gives signed sentences, that do not consider the syntactic structure of both languages and is the most common in literature [1, 24, 57].

Nevertheless, it has some challenges for translation. Oral languages usually have much more words than signs in sign languages; then, words are represented by several signs. Two main cases are compound signs and lexical-visual paraphrases. The first one joins two or more signs to express a concept, for example, the word weekend uses the signs saturday+sunday. The second one describes the concept by several signs, for example, the word burrow is represented by the signs: hole+exactly+home+rabbit.

Syntactic Component. Translation at the morphological level gives signed sentences. That is a word-to-word translation or direct translation. Those word sequences need to be arranged considering the grammatical order of each language.

Semantic Component. Spoken languages have a temporal dimension, they are linear, but in gestural sign languages, the expression is based on two coordinates: space and time, where the spatial dimension is dominant. In addition to the phonemes or minimal signifying units; there are kinetic formative parameters that are the articulatory elements that make up the gestural sign with distinctive value. For example, raising eyebrows to denote causality.

There are other deictic elements as a point of reference with elements such as “this”, “there”, or “now”. Moreover, the dominance of the spatial dimension allows the signer to “place” people or things in space and then use them by modifying the direction of the signs (kineprosema). For example, if the phrase is: “The antenna sends a signal to the cell phone”, the signer first makes the “antenna” sign and “places it spatially” to the right at the top, secondly he makes the “cell phone” sign ” and places it to his left, then he does the sign of “signal” and moves it from right to left joining the first two invisible elements that were placed.

Something similar happens when stories are told, the signer places the interlocutors and turns his back so that he takes the role of one or the other to demonstrate the orientation of the communication between the interlocutors. Reaching this level of translation between two spoken languages is difficult; and even more so when we have sign languages.

5.2 Machine Translation at the Syntactic Level in Sign Language

Machine translation is the use of the computer to realize automatic translation between different languages, from a source language to the target language. It includes data mining and cleansing, word segmentation, part-of-speech tagging, and syntactic analysis [20]. There are two main types of machine translation: rule-based machine translation and corpus-based machine translation (see Section 2.2).

Table 2 shows the studies analyzed in this review. The following sections uses this classification to describe those works. Sign Language Translation requires finding a mapping between a spoken and signed language, that takes into account both their language models, which correspond to the syntactic level.

There are several successful machine translation systems implementing NLP but sign language machine translation has not been widely explored. 28 papers were analyzed that consider the syntactic component of language. Most of the works focus on American Sign Language, also the works using German Sign Language have increased because of the publication of a corpus [8].

Followed by Spanish, Arabic, Spanish and Mexican Sign Language. Other sign languages included in this review are: Swiss German Sign Language, British Sign Language, Pakistan Sign Language, Indian Sign Language, Taiwan Sign Language, Thai Sign Language, Vietnamese Sign Language, Chinese Vietnamese, Ukrainian Sign Language, Portuguese Vietnamese and Italian Sign Language.

Rule-Based: Interlingua and Transfer-Based Machine Translation. The rule-based machine translation includes direct-based, interlingua-based, and transfer-based. Once we focus on the syntactic level, we do not consider the direct-based translation for this study. As it has been said, the problem is not simply mapping text to gestures word-by-word. Most of the rule-based studies focused on rigorous analysis of the grammar of the sign language to define the translation rules, because usually sign languages do not have a formal definition on their countries.

A previous stage of rule-base translation is pre-processing, which allows to prepare the text before the translation stage using tokenization, lemma extraction, and tagging, among others. Some works tokenize the text into words [4, 2], n-grams [44], or sentences [11]. The syntactic analyzer [49, 4, 46, 44] identify the syntactic components of a sentence, such as subject and object. Sign languages, typically, do not consider some syntactic components like prepositions, conjunctions and others so they have to be eliminated as a pre-stage [37, 45] or at the moment of translation [4, 30].

For rule-based translation, some authors just reorder the syntactic components [11, 46, 2, 19], most of the authors did a deep search or analysis of the Sing Language, and obtain a sequence of rules to transform the text into gloss, mainly grammar conversion rules [49, 4, 45, 13, 37, 25, 26, 56]. Other works create an intermediate representation [58, 12, 17], the intermediate representation could include ontology like the semantic ontology of [30] or syntactic trees like [60] with their Synchronous Tree Adjoining Grammar (STAG), and [44, 29] using syntactic trees.

Corpus-Based: Statistical and Hybrid Machine Translation. The corpus-based machine translation could be classified as statistical, example-based, hybrid, or neural. Corpus-based mainly use data sets; Because of the lack of data sets on sign languages, there are fewer studies.

The next section deals with Neural Machine Translation, while in this section we cover the other classes. Wu et al. [59] transform the sentences to possible phrases structure trees from two sets of probabilistic context-free grammars with their own rules. Another work that generates its own grammatical corpus is [39], they build an artificial corpus using grammatical dependencies rules, which is used as input of a statistical machine translation.

Stein, D., Bungeroth, J., and Ney, H. [53] uses phrase-based statistical machine learning on a new corpus of weather reports enhanced by pre and post-processing steps based on the morpho-syntactical analysis of German. Hybrid Machine Translation is uses by [50], they combine statistical translation with an example-based strategy and a rule-based translation method.

Corpus-Based: Neural Machine Translation. Neural machine translation (NMT) is a newly emerging approach to machine translation [23, 28]. The models proposed recently for NMT often belong to a family of encoder-decoders and consist of an encoder that encodes a source sentence into a fixed-length vector from which a decoder generates a translation.

ATLASLang [6], an automatic translation system from Arabic text to Arabic Sign Language (ArSL), uses a backpropagation neural network and focuses on translating simple sentences made up of a limited number of words. A database of signs and morphological characteristics was used to improve the translation. A novel transformer-based architecture is proposed in [9], which jointly learns Continuous Sign Language Recognition and Translation in an end-to-end manner.

By using a Connectionist Temporal Classification (CTC) loss, the recognition and translation problems are combined into a unified architecture, without requiring ground-truth timing information. The approach achieves significant performance gains and outperforms previous translation models on the PHOENIX14T dataset. From the same teamwork, other proposals have been developed.

In [55, 54] they use Neural Machine Translation and Image Generation techniques within three key stages: Text-to-Gloss Neural Machine Translation (NMT) Network, Gloss-to-Motion Lookup Table, and Pose-Conditioned Sign Generation Network. The first one considers the syntactic component; it employs an RNN-based machine translation method using an encoder-decoder architecture with Luong attention for translating spoken language sentences to sign glosses.

6 Challenges and Future Directions

The scarcity of data in sign language translation poses significant challenges due to the vast diversity of sign languages, their lack of standardization, and the limited attention given to deaf individuals. Acquiring extensive datasets is costly and time-consuming, prompting the exploration of alternative approaches that can work effectively with reduced datasets. To address the limitations of Deep Learning models with limited training data, Few-Shot Learning emerges as an approach to learn underlying patterns with just a few training samples. This offers a less expensive solution compared to training large-scale Deep Learning models, which require substantial computational resources and time [41].

Long Language Models (LLMs), such as the LLM GPT-3.5, have demonstrated remarkable capabilities in Natural Language Processing (NLP) tasks, including translation. Scaling up LLMs has shown to greatly enhance task-agnostic, few-shot performance, sometimes outperforming prior state-of-the-art fine-tuning methods [7]. Utilizing the OpenAI API, the LLM GPT-3.5 can be employed and fine-tuned with specific datasets [38], even for tasks like translating from spoken language to sign language gloss. This approach eliminates the need for excessively large datasets, making it a viable option for effective translation.

7 Conclusion

Text to Sign Language Translation has been a widely researched area among various communities worldwide working for the betterment of deaf societies. After reviewing more than 100 articles we have selected 33 published studies. The papers were classified according to different types of machine translation systems, sign language generation methods, and evaluation metrics used. The approach of the present work considerably reduces the articles included since only those that developed a translation that included artificial intelligence techniques and that considered an integral translation at the syntactic level not with signed languages were considered.

References

1. Agrawal, T., Urolagin, S.: 2-way arabic sign language translator using CNNLSTM architecture and NLP. In: Proceedings of the 2020 2nd International Conference on Big Data Engineering and Technology (2020) doi: 10.1145/3378904.3378915
2. Almeida, I., Coheur, L., Candeias, S.: Coupling natural language processing and animation synthesis in portuguese sign language translation. In: Proceedings of the Fourth Workshop on Vision and Language, pp. 94–103 (2015) doi: 10.18653/v1/w15-2815
3. Amazon Web Services: What is machine translation? (2023) aws.amazon.com/what-is/machine-translation/
4. Baldassarri, S., Cerezo, E., Royo-Santas, F.: Automatic translation system to spanish sign language with a virtual interpreter. In: International Federation of Information Processing Conference on Human-Computer Interaction, pp. 196–199 (2009) doi: 10.1007/978-3-642-03655-2_23
5. Bar-Hillel, Y.: The present status of automatic translation of languages. Advances in Computers, vol. 1, pp. 91–163 (1960) doi: 10.1016/s0065-2458(08)60607-5
6. Brour, M., Benabbou, A.: Atlaslang NMT: Arabic text language into arabic sign language neural machine translation. Journal of King Saud University - Computer and Information Sciences, vol. 33, no. 9, pp. 1121–1131 (2021) doi: 10.1016/j.jksuci.2019.07.006
7. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., et al.: Language models are few-shot learners. In: Proceedigns of the Conference on Neural Information Processing Systems, Advances in Neural Information Processing Systems (2020) doi: 10.48550/ARXIV.2005.14165
8. Camgoz, N. C., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7784–7793 (2018) doi: 10.1109/cvpr.2018.00812

9. Camgoz, N. C., Koller, O., Hadfield, S., Bowden, R.: Sign language transformers: Joint end-to-end sign language recognition and translation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10020–10030 (2020) doi: 10.1109/cvpr42600.2020.01004
10. Caselli, N. K., Sehyr, Z. S., Cohen-Goldberg, A. M., Emmorey, K.: ASL-LEX: A lexical database of american sign language. Behavior Research Methods, vol. 49, no. 2, pp. 784–801 (2016) doi: 10.3758/s13428-016-0742-0
11. Dangsaart, S., Naruedomkul, K., Cercone, N., Sirinaovakul, B.: Intelligent thai text – Thai sign translation for language learning. Computers and Education, vol. 51, no. 3, pp. 1125–1141 (2008) doi: 10.1016/j.compedu.2007.11.008
12. Davydov, M., Lozynska, O.: Mathematical method of translation into ukrainian sign language based on ontologies. In: Conference on Computer Science and Information Technologies. Advances in Intelligent Systems and Computing II, pp. 89–100 (2017) doi: 10.1007/978-3-319-70581-1_7
13. El-Gayyar, M. M., Ibrahim, A. S., Wahed, M.: Translation from arabic speech to arabic sign language based on cloud computing. Egyptian Informatics Journal, vol. 17, no. 3, pp. 295–303 (2016) doi: 10.1016/j.eij.2016.04.001
14. Elakkiya, R., Natarajan, B.: ISL-CSLTR: Indian sign language dataset for continuous sign language translation and recognition. Mendeley Data (2021) doi: 10.17632/KCMPDXKY7P.1
15. Farooq, U., Rahim, M. S. M., Sabir, N., Hussain, A., Abid, A.: Advances in machine translation for sign language: Approaches, limitations, and challenges. Neural Computing and Applications, vol. 33, no. 21, pp. 14357–14399 (2021) doi: 10.1007/s00521-021-06079-3
16. Forster, J., Schmidt, C. A., Hoyoux, T., Koller, O., Zelle, U., Piater, J. H., Ney, H.: RWTH-PHOENIX-Weather: A large vocabulary sign language recognition and translation corpus. In: International Conference on Language Resources and Evaluation, pp. 3785–3789 (2012)
17. Grieve-Smith, A. B.: English to american sign language machine translation of weather reports. In: Proceedings of the Second High Desert Student Conference in Linguistics, pp. 23–30 (1999)
18. Gutierrez-Sigut, E., Costello, B., Baus, C., Carreiras, M.: LSE-sign: A lexical database for spanish sign language. Behavior Research Methods, vol. 48, no. 1, pp. 123–137 (2015) doi: 10.3758/s13428-014-0560-1
19. Hernández-Cruz, J.: Traducción de texto en español a texto LSM usando aprendizaje profundo. Tesis de Posgrado, Repositorio Institucional del Tecnológico Nacional de México (2019)
20. Jiang, K., Lu, X.: Natural language processing and its applications in machine translation: a diachronic review. In: IEEE 3rd International Conference of Safe Production and Informatization, pp. 210–214 (2020) doi: 10.1109/iicspi51290.2020.9332458
21. Joshi, A., Agrawal, S., Modi, A.: ISLTranslate: Dataset for translating indian sign language. arXiv (2023) doi: 10.48550/ARXIV.2307.05440
22. Kahlon, N. K., Singh, W.: Machine translation from text to sign language: A systematic review. Universal Access in the Information Society, vol. 22, no. 1, pp. 1–35 (2021) doi: 10.1007/s10209-021-00823-1
23. Kalchbrenner, N., Blunsom, P.: Recurrent convolutional neural networks for discourse compositionality. In: Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality, pp. 119–126 (2013)
24. Kamata, K., Yoshida, T., Watanabe, M., Usui, Y.: An approach to japanese sign language translation system. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 1089–1090 (1989) doi: 10.1109/icsmc.1989.71466

25. Kang, Z.: Spoken language to sign language translation system based on HamNoSys. In: Proceedings of the International Symposium on Signal Processing Systems, pp. 159–164 (2019) doi: 10.1145/3364908.3365300
26. Khan, N. S., Abid, A., Abid, K.: A novel natural language processing (NLP)–based machine translation model for english to pakistan sign language translation. Cognitive Computation, vol. 12, no. 4, pp. 748–765 (2020) doi: 10.1007/s12559-020-09731-7
27. Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A., Corchado, J. M.: Deepsign: Sign language detection and recognition using deep learning. Electronics, vol. 11, no. 11, pp. 1780 (2022) doi: 10.3390/electronics11111780
28. Luong, T., Sutskever, I., Le, Q., Vinyals, O., Zaremba, W.: Addressing the rare word problem in neural machine translation. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, vol. 1, pp. 11–19 (2014)
29. Luqman, H., Mahmoud, S. A.: Automatic translation of arabic text-to-arabic sign language. Universal Access in the Information Society, vol. 18, no. 4, pp. 939–951 (2018) doi: 10.1007/s10209-018-0622-8
30. Mazzei, A., Lesmo, L., Battaglino, C., Vendrame, M., Bucciarelli, M.: Deep natural language processing for italian sign language translation. In: Congress of the Italian Association for Artificial Intelligence. AI*IA 2013: Advances in Artificial Intelligence, pp. 193–204 (2013) doi: 10.1007/978-3-319-03524-6_17
31. Mejía-Peréz, K., Córdova-Esparza, D., Terven, J., Herrera-Navarro, A., García-Ramírez, T., Ramírez-Pedraza, A.: Automatic recognition of mexican sign language using a depth camera and recurrent neural networks. Applied Sciences, vol. 12, no. 11, pp. 5523 (2022) doi: 10.3390/app12115523
32. Moreno-Manzano, D.: English to ASL translator for speech2signs. Universitat Politècnica de Catalunya. Image Processing Group, Signal Theory and Communications Department (2018)
33. Moryossef, A., Yin, K., Neubig, G., Goldberg, Y.: Data augmentation for sign language gloss translation. In: Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages, pp. 1–11 (2021) doi: 10.48550/ARXIV.2105.07476
34. Munib, Q., Habeeb, M., Takruri, B., Al-Malik, H. A.: American sign language (ASL) recognition based on hough transform and neural networks. Expert Systems with Applications, vol. 32, no. 1, pp. 24–37 (2007) doi: 10.1016/j.eswa.2005.11.018
35. Natarajan, B., Elakkiya, R.: Dynamic GAN for high-quality sign language video generation from skeletal poses using generative adversarial networks. Soft Computing, vol. 26, no. 23, pp. 13153–13175 (2022) doi: 10.1007/s00500-022-07014-x
36. Nendauni, L. R.: The development of sign language: A synopsis overview (2021) doi: 10.13140/RG.2.2.19207.93609
37. Nguyen, T. B. D., Phung, T., Vu, T.: A rule-based method for text shortening in vietnamese sign language translation. Information Systems Design and Intelligent Applications, pp. 655–662 (2018) doi: 10.1007/978-981-10-7512-4_65
38. OpenAI: OpenAI: 'Fine-tunes' (2023) platform.openai.com/docs/api-reference/fine-tunes
39. Othman, A., Jemni, M.: Designing high accuracy statistical machine translation for sign language using parallel corpus: Case study english and american sign language. Journal of Information Technology Research, vol. 12, no. 2, pp. 134–158 (2019) doi: 10.4018/jitr.2019040108
40. Othman, A., Jemni, M.: English-ASL gloss parallel corpus 2012: ASLG-PC12. In: Proceedings of the 8th International Conference on Language Resources and Evaluation and 5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, pp. 151–154 (2012)
41. Parnami, A., Lee, M.: Learning from few examples: A summary of approaches to few-shot learning. arXiv (2022) doi: 10.48550/ARXIV.2203.04291

42. Pichardo-Lagunas, O., Martinez-Seis, B.: Multimedia corpora of mexican sign language (MSL) with syntactic functions (2018)
43. Pichardo-Lagunas, O., Martínez-Seis, B., Ponce-de-León-Chávez, A., Pegueros-Denis, C., Muñoz-Guerrero, R.: Linguistic restrictions in automatic translation from written spanish to mexican sign language. In: Mexican International Conference on Artificial Intelligence, Advances in Computational Intelligence, pp. 92–104 (2017) doi: 10.1007/978-3-319-62434-1_8
44. Pichardo-Lagunas, O., Martínez-Seis, B., Ponce-de-León-Chávez, A., Pegueros-Denis, C., Muñoz-Guerrero, R.: Linguistic restrictions in automatic translation from written spanish to mexican sign language. In: Proceedings of the Mexican International Conference on Artificial Intelligence. Advances in Computational Intelligence, pp. 92–104 (2016) doi: 10.1007/978-3-319-62434-1_8
45. Pichardo-Lagunas, O., Partida-Terrón, L., Martínez-Seis, B., Alvear-Gallegos, A., Serrano-Olea, R.: Sistema de traducción directa de español a LSM con reglas marcadas. Research in Computing Science, Advances in Natural Language Processing and Computational Linguistics, vol. 115, pp. 29–41 (2016)
46. Porta, J., López-Colino, F., Tejedor, J., Colás, J.: A rule-based translation from written spanish to spanish sign language glosses. Computer Speech and Language, vol. 28, no. 3, pp. 788–811 (2014) doi: 10.1016/j.csl.2013.10.003
47. Rajalakshmi, E., Elakkiya, R., Subramaniyaswamy, V., Alexey, L. P., Mikhail, G., Bakaev, M., Kotecha, K., Gabralla, L. A., Abraham, A.: Multi-semantic discriminative feature learning for sign gesture recognition using hybrid deep neural architecture. IEEE Access, vol. 11, pp. 2226–2238 (2023) doi: 10.1109/access.2022.3233671
48. Rodríguez-González, M. Á.: Lenguaje de signos. Confederación Nacional de Sordos de España (1992)
49. Royo-Santas, F. J., Silvia-Baldassarri, S.: Traductor de español a LSE basado en reglas gramaticales y morfológicas. In: Proceedings of the VIII Congreso Internacional de Interacción Persona Ordenador, pp. 13–22 (2007)
50. San-Segundo, R., Montero, J. M., Córdoba, R., Sama, V., Fernández, F., D'Haro, L. F., López-Ludeña, V., Sánchez, D., García, A.: Design, development and field evaluation of a spanish into sign language translation system. Pattern Analysis and Applications, vol. 15, no. 2, pp. 203–224 (2011) doi: 10.1007/s10044-011-0243-9
51. Saunders, B., Cihan, N. C., Bowden, R.: Adversarial training for multi-channel sign language production. arXiv(2020)
52. Sharma, P., Tulsian, D., Verma, C., Sharma, P., Nancy, N.: Translating speech to indian sign language using natural language processing. Future Internet, vol. 14, no. 9, pp. 253 (2022) doi: 10.3390/fi14090253
53. Stein, D., Bungeroth, J., Ney, H.: Morpho-syntax based statistical methods for automatic sign language translation. In: Proceedings of the 11th Annual conference of the European Association for Machine Translation (2006)
54. Stoll, S., Camgoz, N. C., Hadfield, S., Bowden, R.: Text2sign: Towards sign language production using neural machine translation and generative adversarial networks. International Journal of Computer Vision, vol. 128, no. 4, pp. 891–908 (2020) doi: 10.1007/s11263-019-01281-2
55. Stoll, S., Camgöz, N. C., Hadfield, S., Bowden, R.: Sign language production using neural machine translation and generative adversarial networks. In: Proceedings of the 29th British Machine Vision Conference, British Machine Vision Association, pp. 1–12 (2018)
56. Sugandhi, Kumar, P., Kaur, S.: Sign language generation system based on indian sign language grammar. ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 19, no. 4, pp. 1–26 (2020) doi: 10.1145/3384202

57. Sáfár, É., Marshall, I.: Sign language translation via DRT and HPSG. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, pp. 58–68 (2002) doi: 10.1007/3-540-45715-1_5
58. Sáfár, É., Marshall, I.: The architecture of an english-text-to-sign-languages translation system. Recent Advances in Natural Language Processing, pp. 223–228 (2001)
59. Wu, C., Su, H., Chiu, Y., Lin, C.: Transfer-based statistical translation of taiwanese sign language using PCFG. ACM Transactions on Asian Language Information Processing, vol. 6, no. 1, pp. 1 (2007) doi: 10.1145/1227850.1227851
60. Zhao, L., Kipper, K., Schuler, W., Vogler, C., Badler, N., Palmer, M.: A machine translation system from english to american sign language. In: Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas: Technical Papers, pp. 54–67 (2000) doi: 10.1007/3-540-39965-8_6

ISSN 1870-4069

Stacking Ensemble for Cognitive Impairment and Alzheimer's Disease Classification Using the ADNI Database

Sergio Vega-Guzmán[1], Gerardo Ramírez-Nava[1,2], Mariel Alfaro-Ponce[1,2]

[1] Tecnológico de Monterrey, Escuela de Ingeniería y Ciencias, Mexico

[2] Tecnológico de Monterrey, Instituto de Materiales Avanzados para la Manufactura Sostenible, Mexico

{a01194108, gerardo.j.ramirez, mariel.alfaro}@tec.mx

Abstract. Dementia is a medical condition encompassing a broad spectrum of cognitive impairments, including a progressive decline in cognitive, motor, and memory skills. Although numerous types of dementia have been identified to date, Alzheimer's disease is still the most extensively studied due to its high prevalence and impact on individuals and society. The Alzheimer's Disease Neuroimaging Initiative (ADNI)is a collaborative research effort dedicated to studying Alzheimer's Disease neuropathology. ADNI has collected clinical data through different study phases, such as laboratory analysis, biomarkers, genetic information, brain imaging, volumetric information, cognitive tests, and other clinical measurements. This information allowed to conform a database that has contributed to the development of multiple scientific studies and clinical trials, including those that have implemented machine learning and deep learning algorithms to classify cognitive impairment stages and the severity of dementia symptoms. Stacked ensemble methods are an interesting alternative that fuses the strengths of several classification base models. This approach has provided flexible frameworks for combining multiple models, leveraging their strengths, and thus making more accurate classifications and predictions. This paper reports a stacking ensemble of classic machine-learning models to classify Alzheimer's disease, normal cognition, and mild cognitive impairment. The stacked ensemble comprises three Gradient Boosting Machine, two Extreme Gradient Boosting models, and two Distributed Random Forests that reached an overall accuracy of 86.9% in the classification process.

Keywords: Cognitive impairment, dementia, stacked ensemble.

1 Introduction

Dementia, as a clinical term, encapsulates a broad spectrum of cognitive impairments where an individual progressively deviates from their normative behavioural patterns to

the extent that they can no longer accomplish tasks routinely expected from a person in their respective age group [11]. This neurodegenerative disease manifests itself most commonly as memory loss but can also come as motor function reduction, spatial awareness decline, and general disorientation and confusion. As per recent studies, it is estimated that around 50 million individuals globally are affected by dementia, a number that showcases the threat of this global health concern [17].

The rising prevalence of dementia worldwide is predicted to escalate further in the future, attributed primarily to the consistent increase in the average lifespan and the consequent growth of the elderly population [12]. Numerous types of dementia have been identified to date, each representing unique facets of neurodegenerative pathologies. These include Vascular dementia, Lewy body dementia, Parkinson's disease, and Alzheimer's disease (AD), each with distinct symptomatology and progression patterns [10].

AD, the most prevalent neurodegenerative pathology, accounts for approximately 70 percent of all dementia occurrences. The alarming rate of its incidence, which is said to double every 5 to 10 years, implies that people in age brackets of 65-69, 70-74, 75-79, 80-84 are at a continually increasing risk, with likelihoods of 0.6%, 1.0%, 2.0%, 3.3%, and 8.4%, respectively [5]. It is pertinent to mention that AD often does not begin with severe symptoms. In many cases, the early stages manifest as Mild Cognitive Impairment (MCI), a condition considered a transitional stage between the expected cognitive decline of normal aging and the more serious decline of dementia.

Individuals with MCI often experience noticeable cognitive changes to the people around them and to themselves, but not severe enough to interfere with their daily life or independent function to a concerning point. Despite not all people with MCI developing AD, a significant proportion do, with studies suggesting that MCI patients progress to AD at a rate of approximately 10-15% per year. Therefore, the importance of the MCI denomination lies in its strong correlation with the progression to AD, making its early detection and study crucial for understanding, preventing, and treating this neurodegenerative condition [14].

Several risk factors contributing to Alzheimer's have been identified in scientific literature, including a family history of dementia, a history of head trauma, certain genetic factors, the presence of two X chromosomes, lower education levels, and vascular disease. These factors, in turn, have led to the identification of several biomarkers that have shown to produce accurate classification results when incorporated into machine learning and deep learning algorithms [5].

These algorithms have been trained on clinical and imaging data to produce an acceptable model capable of classifying AD stages or forecasting the progression from MCI to AD. In the relevant literature, several examples of this can be found. For instance, the work of Beltrán [2] used the ADNI database to predict the transition from MCI to AD. To do so, several machine learning models were implemented, with Random Forests (RF) and Gradient Boosting Machines (GBM) being the most successful of them, achieving an AUC of 0.77 in the forecasting task. Similarly, Dimitriadis [8] also used the ADNI database to create a new and unique four-class AD-based problem. By integrating morphological MRI-based features such as cortical thickness, subcortical volumes, and hippocampal subfields within a Random Forest

framework, the study achieved a 61.9% classification performance in distinguishing between four groups: Healthy Control, MCI, converted MCI, and AD. Doyle [9] forecasted the development of AD using multivariate ordinal regression to model the ordered brain deterioration from normal aging (CTL) to MCI to AD. Wang [16] developed a hybrid machine learning system that combines multiple convolutional neural networks and a linear support vector classifier. According to clinical evidence, convolutional neural networks were used to automatically extract image features from brain segments related to cognitive decline.

The linear support vector classifier then used the extracted image features and non-image information to make the final predictions. Recently, stacked ensembles have been successfully implemented in medical diagnostics and a variety of other fields. Stacked ensemble methods have improved the predictive performance of a model by combining the strengths of several base models and feeding their predictions into a higher-level, secondary model (meta-learner) to produce the final prediction.

The primary purpose of this technique is to blend the capabilities of numerous diverse models to mitigate individual model weaknesses, improve generalization, and enhance the overall predictive accuracy [13]. For example, stacked ensemble models have been utilized for neuropathologies to predict AD onset [1] by combining different machine learning algorithms.

In this project, a novel methodology for classifying Alzheimer's disease, normal cognition, and mild cognitive impairment was proposed using a stacking ensemble of classic machine learning models. This paper is structured as follows: Section 2 Methodology describes the database and the clinical data considered in the study, the processing and feature selection of the data. The stacked ensemble model is also reported in this section. Section 3 reports the performance and accuracy of the stacked ensemble. Finally, the conclusions are outlined in the last section of the document.

2 Methodology

The data analysis and model training for this study were carried out on a virtual computer with the following specifications: the operating system was a Linux distribution, the virtual machine architecture was x86_64, the full platform description was Linux-5.15.107+-x86_64-with-glibc2.31, the processor was an x86_64, the total CPU count was 2, and the system was equipped with a total memory of approximately 12.68 GB. Figure 1 depicts the general pipeline of the proposed stacking ensemble algorithm.

2.1 Database

In this project, multiple datasets from the ADNI database [15] were considered. The primary dataset of the study, the ADNIMERGE dataset encapsulates critical information from various phases of the ADNI project (ADNI1, GO, 2, 3) and it comprises 16,345 rows and 42 columns, capturing a broad spectrum of participants information across different stages of the disease.

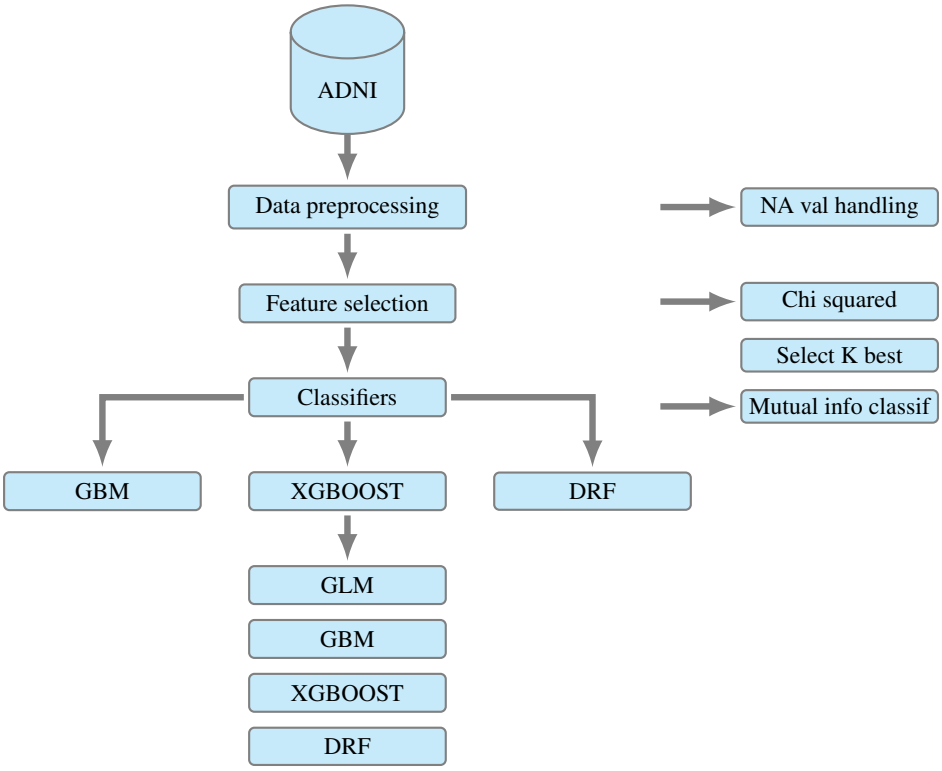


Fig. 1. General pipeline of the proposed stacking ensemble algorithm.

However, this dataset has the disadvantage of having a significant amount of missing data particularly in the 'ABETA,' 'TAU,' and 'PTAU' columns, with over 13,975 missing values each. In order to improve the dataset, an integration of the TOMM40 PolyT Variant Data and the Desikan Lab Polygenic Hazard Score (PHS) was made. The TOMM40 dataset, consisting of 1,520 rows and five columns, provides a clean and focused view of the TOMM40 gene. The PHS dataset, on the other hand, containing 757 rows and five columns, reveals minor data inconsistencies with 11 missing values each in the 'TOMM40_A1' and 'TOMM40_A2' columns.

These columns potentially represent alleles of the TOMM40 gene, enhancing the understanding of the genetic influence on Alzheimer's disease progression. After this process, the unified dataset encompassing approximately 2000 participants categorizes individuals into three cognitive states: Normal cognition, Mild Cognitive Impairment (MCI), and Alzheimer's Disease (AD). This diverse dataset improved the external validity and reliability of the machine learning models.

It is worth mentioning that a significant part of the study is centered around the analysis of Mild Cognitive Impairment (MCI), a transitional stage between normal cognitive aging and dementia, since the analysis of MCI helps identify the early stages of cognitive decline, capturing the subtle yet significant shifts that a person undergoes when they drift away from normal cognition.

The features utilized in this study are categorized into five primary groups: genetic data, protein data, radiopharmaceutical data, brain volumetric data, demographic data, and cognitive assessment data.

2.2 Data Preprocessing

During the initial stages of data preprocessing, a significant amount of missing values in the dataset was identified. This posed a significant problem, especially considering the requirement for developing a classifier that is not reliant on incomplete or artificially augmented data. Upon closer inspection, it was found that certain fields, such as 'FDG,' 'ABETA,' 'TAU,' 'PTAU,' and, at certain stages of the ADNI dataset, the brain volume data, contained fewer complete records compared to other variables.

For the data cleanup stage, which in part involved the elimination of rows with missing data, the loss of two entire classes: 'SMC' and 'EMCI,' was observed. Regarding genetic data, the 'APOE4' column was considered. This captures information about the presence of the APOE4 allele, which has been associated with an increased risk of Alzheimer's disease. In the category of proteic data, the columns 'ABETA,' 'TAU,' and 'PTAU' were included. These columns contain information about various Alzheimer-related proteins, which serve as biochemical markers for the disease.

For radiopharmaceutical data, the 'FDG' column was considered. Regarding imaging data, the columns 'Ventricles,' 'Hippocampus,' 'WholeBrain,' 'Entorhinal,' 'Fusiform,' 'MidTemp,' and 'ICV' were selected. These columns contained volumetric measurements of various brain regions and the overall intracranial volume. The demographic data considered the columns: 'AGE,' 'PTGENDER,' 'PTEDUCAT,' 'PTETHCAT,' 'PTRACCAT,' and 'PTMARRY .' Lastly, for cognitive assessment data, the following columns were included: 'CDRSB,' 'ADAS11', 'ADAS13', 'ADASQ4', 'MMSE,' 'RAVLT_immediate,' 'RAVLT_learning,' 'RA- VLT_forgetting,' 'RAV LT_perc_forgetting,' 'LDELTOTAL,' 'DIGITSCOR,' 'TRABSCOR,' 'FAQ.' and there were no experimental configurations discovered that could retain these two classes without a severe hindrance on the model's performance.

Another issue related to protein data and how it was stored in the CSV files was that the three relevant columns 'ABETA,' 'TAU,' and 'PTAU' had information written as a string when concentrations exceeded or did not reach a certain value. The adjustments performed were simply the swapping of specific string values to their closes numerical representation. In detail, the process was done as follows:

1. The data associated with the 'ABETA' protein was transformed by replacing any instances of values greater than 1700 and less than 200 with 1700 and 200, respectively.
2. Similarly, the 'TAU' protein data was adjusted by modifying the instances of values above 1300 and below 80, with 1300 and 80, respectively.
3. Finally, the data associated with the 'PTAU' protein was updated by replacing the occurrences of values exceeding 120 and falling below 8 with 120 and 8, respectively.

2.3 Feature Selection

For the feature selection stage, the 'SelectKBest' function combined with the 'mutual_info_classif' method, both from Scikit-learn's feature selection module were used. 'SelectKBest' is a univariate feature selection method that identifies the 'k' highest scoring features. The 'mutual_info_classif' method is designed to compute the Mutual Information (MI) between each feature and the target variable, in this case, 'DX_bl'. The mathematical formula [7] underpinning this method is as follows:

$$\mathrm{MI}(X,Y) = \sum_{x}\sum_{y} p(x,y)\log\left(\frac{p(x,y)}{p(x)\,p(y)}\right). \tag{1}$$

In this equation, X represents a feature, and Y symbolizes the target variable. $p(x,\, y)$ is the joint probability distribution function of X and Y, whereas $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y, respectively. Mutual information is beneficial as it measures the dependency between the variables and only returns a zero when two variables are found to be independent.

The relevance of each feature was determined by calculating Mutual Information (MI), which indicated the strength of its relationship with the target variable. These scores were then assessed, focusing especially on categorical and numerical features with non-zero mutual information. This non-zero value signified a degree of correlation with the target variable. This feature selection process allowed for a focus on the most relevant attributes, thereby improving the precision and efficiency of the predictive model.

2.4 Classification Model

The study implemented an ensemble model that utilized a meta learner algorithm based on generalized linear models (GLM) with a logit transformation. The ensemble was constructed using the following base models:

- **GBM_4 Model (Gradient Boosting Machine):**
  - Trained with 48 trees, a maximum depth of 10, and a learning rate of 0.1.
  - Utilized a multinomial distribution and employed a UniformAdaptive histogram type.
- **DRF_1 Model (Distributed Random Forest):**
  - Trained with 32 trees and a maximum depth of 20.
  - Utilized a multinomial distribution and employed a UniformAdaptive histogram type.
- **XGBoost_3 Model (eXtreme Gradient Boosting.):**
  - Trained with 40 trees, a maximum depth of 5, and a learning rate of 0.3.
  - Utilized a multinomial distribution and employed the exact tree method with a depthwise grow policy.

– **GLM_1 Model (Generalized Linear Model):**
  – Trained using a multinomial family with coordinate descent as the solver.
  – Employed lambda search with early stopping.
– **XRT_1 Model (Extremely Randomized Trees, Treated as DRF):**
  – Trained with 26 trees and a maximum depth of 20.
  – Utilized a multinomial distribution and employed a random histogram type.

XGBoost, short for "eXtreme Gradient Boosting," is an optimized implementation of a Gradient Boosting Machine (GBM). XGBoost improves upon the classic GBM framework by introducing regularization to avoid overfitting, as well as several system optimizations to speed up and improve the model's performance. In essence, the XGBoost algorithm works by iteratively adding new models to the ensemble that predict the errors of the previous models. The prediction [6] at each step is given by:

$$\mathcal{F}_m(x_i) = \mathcal{F}_{m-1}(x_i) + \langle_m(x_i), \tag{2}$$

where $\mathcal{F}_m(x_i)$ is the predicted output after the mth model, $\mathcal{F}_{m-1}(x_i)$ is the prediction from the previous step, and $\langle_m(x_i)$ is the current model that's added to improve the prediction by predicting the residuals of the previous model. Distributed Random Forest (DRF) models operate similarly to the standard Random Forest algorithm, with variations in their configuration to ensure diversity among the predictions of individual trees in the ensemble. A Random Forest or DRF model can be abstractly represented [3] as:

$$Y = \frac{1}{n}\sum_{i=1}^{n} T_i(X), \tag{3}$$

where Y is the output variable, X is the vector of input variables, $T_i(X)$ represents the prediction of the $i-th$ decision tree in the ensemble, and n is the total number of trees in the ensemble. The final output of the stacked ensemble model is a weighted sum of the individual model predictions and can be represented [4] as:

$$F(x_i) = \sum_{m=1}^{M} w_m\, F_m(x_i), \tag{4}$$

where, $F(x_i)$ is the final output, $F_m(x_i)$ is the output of the mth model, and w_m is the weight for the mth model. These weights are learned during training to optimize the ensemble's performance.

K-fold cross validation. Is a statistical method used for estimating the performance of predictive models. This type of validation is mostly used when a model's goal is prediction, and one wants to estimate its accuracy with as little bias as possible. This study employed a specific form of cross-validation called 5-fold cross-validation. In this approach, the dataset was divided into five equal-sized folds, and then the predictive model was trained and validated five times, with each iteration using a different fold for validation while the remaining folds were used for training.

Table 1. Feature importance and categories for the ensemble model.

Feature	Score_MI	Category
cdrsb	0.596834	Clinical
ldeltotal	0.439072	Neuropsychological
mmse	0.423307	Clinical
faq	0.422767	Clinical
adas13	0.418954	Clinical
adasq4	0.405963	Clinical
cir	0.345453	Biomarkers
adas11	0.329069	Clinical
phs	0.282552	Biomarkers
ravlt_perc_forgetting	0.261960	Neuropsychological
ravlt_immediate	0.236800	Neuropsychological
ravlt_learning	0.190921	Neuropsychological
fdg	0.187301	Imaging
ptau	0.174670	Biomarkers
trabsor	0.173784	Clinical
ravlt_forgetting	0.157009	Neuropsychological
digitscor	0.140957	Neuropsychological
hippocampus	0.133758	Imaging
apoe4_0.0	0.130713	Genetic
abeta	0.118638	Biomarkers
fusiform	0.115856	Imaging
tau	0.097753	Biomarkers
entorhinal	0.084071	Imaging
midtemp	0.079739	Imaging
tomm40_a1	0.078894	Genetic
apoe4_1.0	0.071801	Genetic
icv	0.059267	Imaging
ptmarry_married	0.053313	Demographic
tomm40_a2	0.051060	Genetic
wholebrain	0.034028	Imaging
ptraccat_white	0.029367	Demographic
ptmarry_widowed	0.024984	Demographic
ventricles	0.023164	Imaging
ptgender_female	0.021457	Demographic
ptethcat_not hisp/latino	0.014322	Demographic
ptmarry_never married	0.002360	Demographic
ptmarry_divorced	0.001434	Demographic

A mathematical representation for the average performance in 5-fold cross-validation can be expressed as:

$$E = \frac{1}{5} \sum_{i=1}^{5} E_i, \tag{5}$$

where E is the average performance across the folds, and E_i is the performance metric.

Table 2. Performance metrics of various classifiers.

Classifier	Mean Accuracy	SD	CV1	CV2	CV3	CV4	CV5
Logistic Regression	0.835486	0.038173	0.773585	0.865385	0.884615	0.826923	0.826923
Random Forest	0.724311	0.046802	0.679245	0.769231	0.788462	0.673077	0.711538
Support Vector Machine	0.853661	0.030135	0.828125	0.831169	0.888889	0.836066	0.884058
Gradient Boosting	0.770174	0.039931	0.754717	0.730769	0.846154	0.769231	0.750000
XGBoost	0.839260	0.034568	0.792453	0.884615	0.846154	0.865385	0.807692
Ensemble Classifier	0.869884	0.021199	0.830189	0.884615	0.884615	0.884615	0.865385

3 Results

3.1 Data Preprocessing

As previously mentioned, the data employed for this study is made of multiple datasets from the ADNI database. In order to be able to work with it, the clean-up of the data plays an important role in the implementation of the staking ensemble. After carrying out the preprocessing, the database went from having 16345 incomplete rows, 42 columns, and five classes (LMCI, CN, AD, EMCI, SMC) to having 411 rows, 36 columns, and three classes (LMCI, CN, AD). Of those additional columns, 4 correspond to integrating the TOMM40 PolyT Variant Data and the Desikan Lab Polygenic Hazard Score (PHS) associated information.

3.2 Feature Selection

As a second step, a feature selection was performed on the 'clean' database to improve the performance of the ML algorithm. According to the results that can be seen in Table 1, cognitive test data (labeled as clinical) was shown to have the highest MI scores, meaning that these features have a substantial impact on the model's predictive accuracy, follow up by neuropsychological data.

Conversely, demographic features demonstrated the lowest MI scores, indicating a lesser contribution to the model's predictions, and although these factors did contribute to some extent, their impact was not as pronounced as that of the cognitive tests. This result is consistent with what is reported by medical specialists, which gave a higher weight to clinical, neuropsychological, imaging, biomarkers, and generic data.

3.3 Classification Model

After the feature selection stage, a stacked ensemble model was implemented; this model ensured a robust prediction method by leveraging the strengths of different machine learning algorithms. For this specific case a GBM, XGBoost, and DRF, combined with a powerful meta-learner (GLM), optimally enhanced these base models predictions. Table 2 shows the performance of the stacked ensemble as a validation method a k-fold cross-validation with $k = 5$ was employed, obtaining an accuracy of 86.9%.

4 Conclusion

This study was done using the ADNI database and its multiple datasets. Two main preprocessing steps were performed due to several data utilization problems. Specifically, a large portion of incomplete rows needed to be eliminated, and several string records in the protein columns needed to be replaced by their closest numerical representation.

In the feature ranking analysis conducted on the preprocessed dataset using the random forest algorithm, cognitive examination data emerged as the most significant predictor for Alzheimer's disease. This was closely followed by indicators such as the presence of the ptau protein and volumetric measurements of the hippocampus, a region notably affected in Alzheimer's pathology.

The pronounced significance of the examination data can be attributed to its direct and intrinsic nature. While various biomarkers and neuroanatomical measurements provide valuable insights into the disease's progression and manifestations, direct cognitive assessments capture the immediate and functional impact of the disease on an individual's cognitive abilities. As such, by their very nature, these examinations are poised to inherently possess greater diagnostic relevance than indirect predictors.

In the experimental phase, several configurations were tested. The most promising results were obtained when all features were considered. The optimal stacking ensemble architecture consisted of seven foundational models: three Gradient Boosting Machines (GBM), two Extreme Gradient Boosting models (XGBoost), and two Distributed Random Forests (DRF). Evaluated using a 5-fold cross-validation method, this model configuration achieved an overall accuracy of 86.9%.

Acknowledgments. The authors acknowledge ADNI for database access; CONAHCYT (grant number 1239365) and Tecnologico de Monterrey for their financial support.

References

1. Alatrany, A. S., Hussain, A., Jamila, M., Al-Jumeiy, D.: Stacked machine learning model for predicting Alzheimer's disease based on genetic data. In: 14th International Conference on Developments in eSystems Engineering, IEEE, pp. 594–598 (2021) doi: 10.1109/DeSE54285.2021.9719449
2. Beltrán, J. F., Wahba, B. M., Hose, N., Shasha, D., Kline, R. P.: Inexpensive, non-invasive biomarkers predict Alzheimer transition using machine learning analysis of the Alzheimer's disease neuroimaging (ADNI) database. Public Library of Science ONE, vol. 15, no. 7 (2020) doi: 10.1371/journal.pone.0235663
3. Breiman, L.: Random forests. Machine Learning, vol. 45, no. 1, pp. 5–32 (2001) doi: 10.1023/A:1010933404324
4. Brownlee, J.: How to develop a weighted average ensemble with python. Ensemble Learning (2021) machinelearningmastery.com/weighted-average-ensemble-with-python/
5. Castellani, R. J., Rolston, R. K., Smith, M. A.: Alzheimer disease. Disease-a-Month, vol. 56, no. 9, pp. 484–546 (2010) doi: 10.1016/j.disamonth.2010.06.001

6. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In: Proceedings of the 22nd Association for Computing Machinery, Special Interest Group on Knowledge Discovery in Data, International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016) doi: 10.1145/2939672.2939785
7. Cover, T. M., Thomas, J. A.: Elements of information theory. John Wiley and Sons, Inc (2005) doi: 10.1002/047174882x
8. Dimitriadis, S. I., Liparas, D., Tsolaki, M. N.: Random forest feature selection, fusion and ensemble strategy: Combining multiple morphological MRI measures to discriminate among healhy elderly, MCI, cMCI and Alzheimer's disease patients: From the Alzheimer's disease neuroimaging initiative (ADNI) database. Journal of Neuroscience Methods, vol. 302, pp. 14–23 (2018) doi: 10.1016/j.jneumeth.2017.12.010
9. Doyle, O. M., Westman, E., Marquand, A. F., Mecocci, P., Vellas, B., Tsolaki, M., Kłoszewska, I., Soininen, H., Lovestone, S., Williams, S. C. R., Simmons, A.: Predicting progression of Alzheimer's disease using ordinal regression. Public Library of Science ONE, vol. 9, no. 8 (2014) doi: 10.1371/journal.pone.0105542
10. Gauthier, S., Reisberg, B., Zaudig, M., Petersen, R. C., Ritchie, K., Broich, K., Belleville, S., Brodaty, H., Bennett, D., Chertkow, H., Cummings, J. L., de-Leon, M., Feldman, H., Ganguli, M., Hampel, H., Scheltens, P., Tierney, M. C., Whitehouse, P., Winblad, B.: Mild cognitive impairment. The Lancet, vol. 367, no. 9518, pp. 1262–1270 (2006) doi: 10.1016/s0140-6736(06)68542-5
11. Geldmacher, D. S., Whitehouse, P. J.: Evaluation of dementia. New England Journal of Medicine, vol. 335, no. 5, pp. 330–336 (1996) doi: 10.1056/NEJM199608013350507
12. Lee, M., Chodosh, J.: Dementia and life expectancy: What do we know? Journal of the American Medical Directors Association, vol. 10, no. 7, pp. 466–471 (2009) doi: 10.1016/j.jamda.2009.03.014
13. Pavlyshenko, B.: Using stacking approaches for machine learning models. In: IEEE Second International Conference on Data Stream Mining and Processing, pp. 255–258 (2018) doi: 10.1109/DSMP.2018.8478522
14. Petersen, R. C.: Mild cognitive impairment. CONTINUUM: Lifelong Learning in Neurology, vol. 22, no. 2, pp. 404–418 (2016) doi: 10.1212/con.0000000000000313
15. University of Southern California: ADNI database. image and data archive laboratory of neuro imaging (2023) ida.loni.usc.edu/
16. Wang, C., Li, Y., Tsuboshita, Y., Sakurai, T., Goto, T., Yamaguchi, H., Yamashita, Y., Sekiguchi, A., Tachimori, H.: A high-generalizability machine learning framework for predicting the progression of Alzheimer's disease using limited data. npj Digital Medicine, vol. 5, no. 1, pp. 43 (2022) doi: 10.1038/s41746-022-00577-x
17. World Health Organization: Risk reduction of cognitive decline and dementia: WHO guidelines (2019) www.who.int/publications/i/item/9789241550543

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación
en Computación