EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

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

Research in Computing Science

Vol. 153 No. 12
December 2024

Research in Computing Science

Research in Computing Science, Año 23, Volumen 153, No. 12, diciembre de 2024, 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 diciembre de 2024.

Research in Computing Science, year 23, Volume 153, No. 12, December 2024, is published monthly by the Center for Computing Research of IPN.

Advances in Artificial Intelligence

Gilberto Ochoa-Ruiz (ed.)

Instituto Politécnico Nacional, Centro de Investigación en Computación
México 2024

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

ISSN 1870-4069

Detection and Classification of Brain Tumors in MRIs Using CNNs

Brandon Alejandro Llaca-Sánchez, Dulce Sofía Marín-Chávez, Alejandra Vilchis-Yubi

Universidad Autónoma de Querétaro,
Facultad de Ingeniería, Campus Aeropuerto, Santiago de Querétaro, Mexico

{bralig14, dulcesofia021098, alevyubi}@gmail.com

Abstract. This paper describes implementing a Convolutional Neural Network (CNN) model for detecting and classifying brain tumors from Magnetic Resonance Images (MRIs) using TensorFlow and Keras. The model classifies images into one of 4 possible classes: glioma, meningioma, pituitary, or "no tumor" (healthy patient). The data augmentation paradigm was used alongside image processing techniques to expand the training dataset, achieving a final test set accuracy of 89%. Brain tumors present challenges in their detection and classification due to their variability in shape, size, and location, which complicates medical diagnosis using traditional methods promptly. To address this challenge, this study employs a CNN model that integrates convolutional layers alternated with pooling layers, inspired by modifications of existing architectures that have proven to be efficient in terms of the computational cost-accuracy ratio. This work aims to refine the accuracy of classification among different types of tumor, but also versus non-tumor images. Furthermore, a user-friendly Python-based graphical interface has been developed to enable users unfamiliar with deep learning models to conduct preliminary MRI classifications, potentially saving diagnostic time and resources in medical environments.

Keywords: Brain tumor, convolutional neural network, MRIs, data augmentation, graphical interface.

1 Introduction

This paper applies a deep learning model (using Convolutional Neuronal Networks, CNNs) with data augmentation and image processing techniques, to detect and classify brain tumors in magnetic resonance images, using Tensorflow and Keras. Brain tumors pose a challenge in their detection and classification due to their variability in shape, size, and location, making timely medical diagnosis difficult using conventional techniques. In response to this problem, a convolutional neural network-based model inspired by modifications of existing architectures was used in this study and was shown to be efficient in terms of computational cost/accuracy ratio to improve the accuracy of classification between tumor types and non-tumor cases.

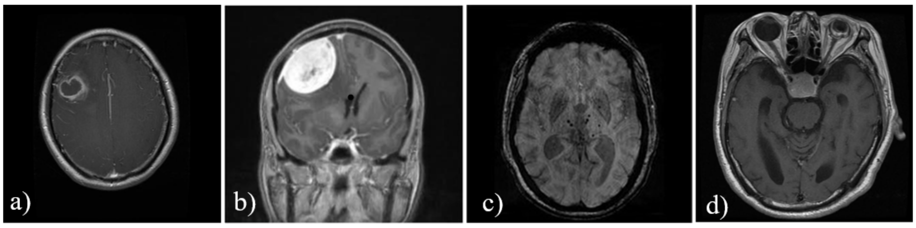


Fig. 1. Examples of MRIs of each class, a) Glioma, b) Meningioma, c) No tumor, d) Pituitary.

To balance the variability of clinical images and avoid the need for more data, data augmentation techniques such as geometric transformations and intensity adjustments are applied. The transformations include partial rotations, inversions, and contrast adjustments, which give the model the ability to identify tumors under different visual conditions and increase its robustness to images from new patients.

TensorFlow was used to create and train CNN models tailored to the specific characteristics of MRIs. The networks were trained with an extended dataset containing images of gliomas, meningiomas, pituitary tumors, and healthy patients, all normalized to a uniform format of 250×250 pixels to standardize the inputs to the model. The present research focuses on improving the accuracy of brain tumor classification using CNNs and data augmentation techniques. The results obtained could then be a useful tool for medical professionals to improve the efficiency of brain tumor diagnosis.

2 Related Work

There are many research papers on the classification of brain tumors. Some of them are Mohsen [10], who used a Deep Neural Network Classifier with a dataset of 66 classes of brain tumors (glioblastoma, sarcoma, and metastatic bronchogenic carcinoma) and normal MRIs using a discrete wavelet transform (DWT) and principal component analysis (PCA) with a classification rate of over 93%. Rai [13] presents a deep neural network called U-NEt (LU-Net), which distinguishes between normal and abnormal MRI images of the brain using a data set of 253 images and achieves an accuracy of 88%. A Convolutional Neural Network with a Long Short Term Memory (LSTM) was used by the authors in [16] for feature extraction to augment the CNN extraction features with a dataset of 3264 MRI scans.

Nyoman Abiwinanda [6] trained different CNN architectures such as AlexNet, ResNet, and VGG16 to detect 3 types of brain tumors (glioma, meningioma, and pituitary). The dataset used consists of 3064 T-1 weighted CE-MRI images. The best-performing architecture had a training accuracy of 98.51% and a validation accuracy of 84.19%. Yakub Bhanothu [4] proposed an algorithm called "Faster R-CNN", which is also used to classify and detect the 3 types of brain tumors mentioned above. The algorithm uses VGG-16 as the base layer and consists of three blocks called RPN, a region of interest (RoI) pooling and a regional-based convolutional neural network. The dataset used is public, it contained 805 MRIs for glioma, 694 for meningioma, and 907 for tumors. They achieved an average precision of 77.60%.

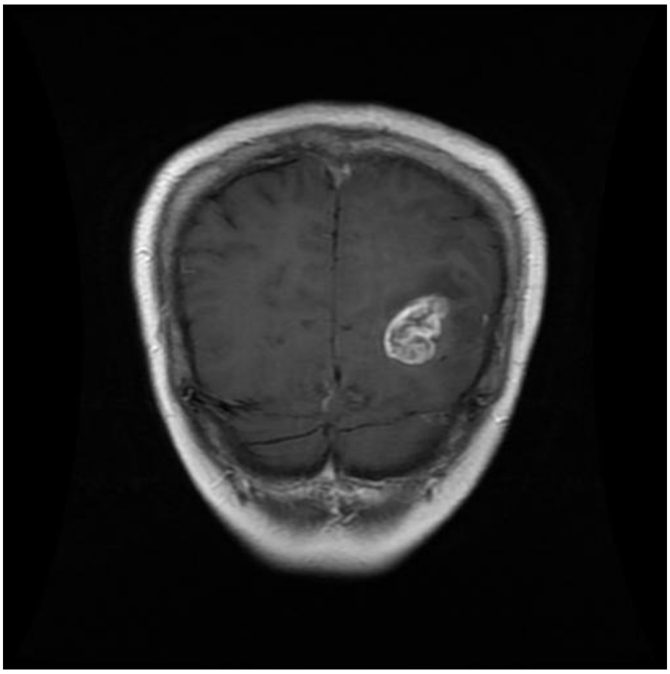


Fig. 2. MRIs of Glioma tumor example.

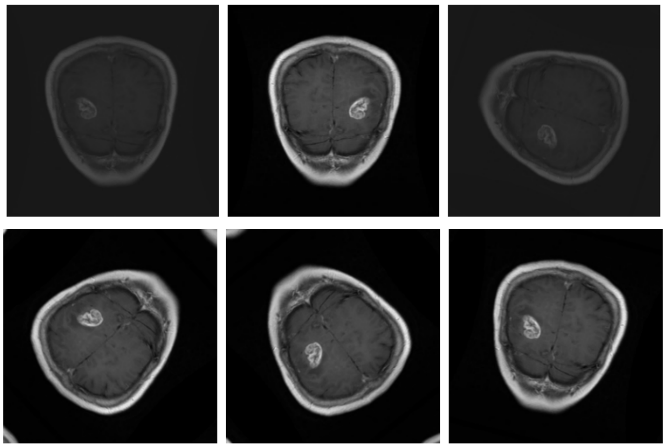


Fig. 3. Example of the 6 transformations applied to the side glioma image.

An interesting approach was taken by the authors in [2], where they first used several filters to preprocess the MRIs, aiming to differentiate the various elements of the image more easily. This preprocessing step facilitated the clustering and segmentation of tumors. A prediction of tumor versus non-tumor was then made using a stacked sparse autoencoder (SSAE) model with two fine-tuned layers, obtaining an accuracy between 90-100% on the BRATS (Brain Tumor Segmentation) datasets [3].

```
Size of x_train: (6720, 250, 250, 3)
Size of y_train: (6720,)
Size of x_test: (120, 250, 250, 3)
Size of y_test: (120,)
Size of x_val: (120, 250, 250, 3)
Size of y_val: (120,)
```

Fig. 4. Sizes and dimensions of training, validation, and test sets.

Regarding data augmentation methods, in [8] Han et al. leveraged Progressive Growing of GANs (PGGANs) to generate synthetic 256x256 MRIs. This method, which was evaluated on the BRATS 2016 dataset [9] and combined with traditional data augmentation methods, improved brain tumor detection accuracy to 91.08%, with 86.60% of sensitivity and 97.60% of specificity, delivering promising results for clinical use. The main goal of this research, which focuses on detecting and classifying brain tumors, is to apply these models in software or interfaces that are user-friendly for the medical staff. Ucuzal [15] developed a free web-based on deep learning that can be used in the detection of brain tumors, specifically Glioma, Meningioma, and Pituitary tumors on MRIs. They used a Keras library to build a deep learning model and achieved 95% accuracy in the test dataset.

However, for further studies, they propose to include the classification and detection of MRIs of healthy patients [15]. An extensive review of more related works in the field (until 2021) is provided in [11]. Al-Zoghby, who proposed the DCTN model using a CNN with VGG-16 architecture and a dataset with 233 patients, achieved 99% accuracy [1] during testing. Özkaraca used DenseNet, VGG-16, and modified CNN architectures, reaching an accuracy of up to 92%.

However, in the confusion matrix for the VGG-16 model, the values obtained showed many mistakes in glioma classification [12]. Srinivasan presented a comparison of AlexNet, DenseNet121, ResNet-101, VGG-19, and GoogleNet models, reaffirming the superiority of CNN in the field of brain tumor classification [14]. As reviewed by many authors, significant results have been achieved in detecting and classifying brain tumors using CNNs. Two of the main contributions of this investigation are to implement and continue exploring the data augmentation approach for this problem, as well as to develop a graphical Python interface that is easy for medical staff to use and understand, in order to assist them in making fast and appropriate diagnoses.

3 Methods

The methodology of this study involved various phases, from data preparation to the training and evaluation of the neural network. The steps taken to implement the model successfully are described below:

3.1 Dataset

The study was performed using the public database ”Brain MRI Scans for Brain Tumor Classification” obtained from the platform ”Kaggle” [7]. The dataset contains

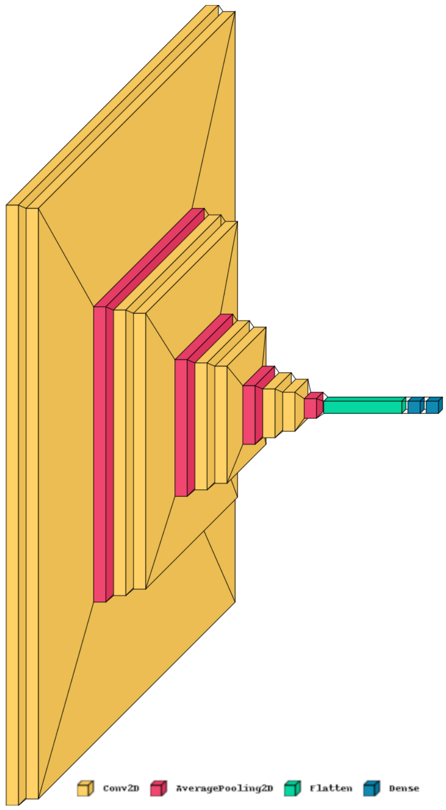


Fig. 5. Three-dimensional representation of the network architecture. Yellow layers: Convolutional layers; Red layers: Pooling layers; Green layer: Flatten layer; Blue layers: Dense layers.

high-quality Brain Magnetic Resonance Imaging scans with diverse tumor types. It is classified into four classes and each image is labeled with one of these classes: “Pituitary”, “Glioma”, “Meningioma” and “No tumor”. It has a total number of images of 1311, where 300 images belong to Pituitary, 306 images to Meningioma, 300 images to Glioma, and 405 images to No tumor class. Fig. 1 shows an example of MRIs that belong to each class.

3.2 Data Processing

To achieve a balanced distribution of classes, it was first decided to select the first 300 available images of each class to homogenize the number of data per class, since Meningioma and No tumor classes have more than 300 images.

All images were uploaded to the working environment, 300 images for each class. The image set for each class was divided into 240 images for training about 30 images for testing and 30 for validation. The selection of images for each group was randomized. The total number of images for the training process was 960 images, and

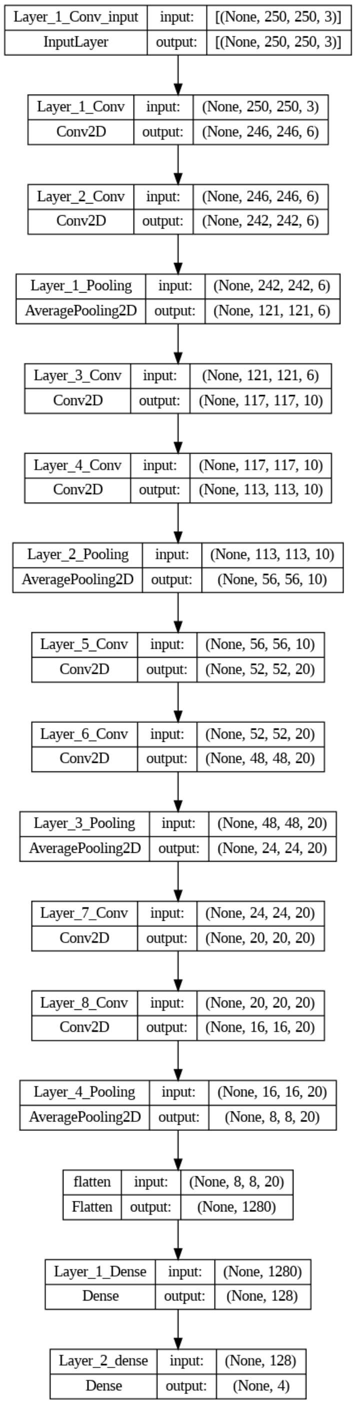


Fig. 6. Two-dimensional representation of network parameters and network architecture.

```
Epoch 20/20
105/105 [==============================] - 8s 81ms/step -
```

```
loss: 0.1162 - accuracy: 0.9585 - val_loss: 0.7878 - val_accuracy: 0.8083
```

Fig. 7. Metrics achieved in 20 epochs of training.

120 images each for the testing and validation process. The data augmentation process was performed on the 960 training images. The data augmentation was performed with TensorFlow. The transformations used were the change of image contrast, horizontal and vertical flips, and partial rotations.

Resizing was also performed, to obtain the same size for all data and to standardize the network inputs, the images were resized to 250×250 pixels. For each original image, 6 new images were generated by data augmentation. The number of generated images was chosen due to the limited amount of original images available for training and the limited computational power, resulting in a good balance that proved to be efficient considering the results, with the transformation applied either directly to the original image or to a previously generated image.

Fig. 2 shows that the original image belongs to the glioma class and Fig. 3 shows an example of the 6 transformations applied to that image, which, as mentioned previously, were a combination of changes in image contrast, horizontal and vertical flips, and partial rotations. A total of 6720 images were generated for model training, in which each class consisted of 1680 images. The final dimensions and sizes of the training, test, and validation sets are shown in Fig. 4.

3.3 Model Architecture

To reduce dimensionality, extract features and avoid overfitting, a convolutional network architecture with alternating convolutional layers and pooling layers was chosen. Fig. 5 shows a three-dimensional representation of the network architecture. The architecture of the model can be seen systematically in Fig. 6, which shows the layers that make up the CNN model. It consists of eight convolutional layers, four pooling layers, one fully connected layer, and one dense layer. Each pair of convolutional layers contains 6, 10, 20, and 20 kernels, all of size 5×5, and each layer also contains a ReLu activation function. In contrast, the size of the pooling layers is 2×2. The first convolutional layer receives input data of size 250×250. The dense layers at the end are used for classification into the four diagnostic categories.

3.4 Model Compilation and Training

The model was created with the Adam optimizer and a learning rate of 0.001, using categorical cross entropy as the loss function. It was trained for 20 epochs with batches of 64 images, with hyperparameters adjusted based on performance during validation.

```
Test Accuracy: 89.17%
Test Loss: 0.5614
```

Fig. 8. Metrics achieved on testing.

	precision	recall	f1-score	support
0	0.77	0.89	0.83	27
1	0.87	0.76	0.81	34
2	0.97	0.97	0.97	32
3	0.96	0.96	0.96	27
accuracy			0.89	120
macro avg	0.89	0.90	0.89	120
weighted avg	0.89	0.89	0.89	120

Fig. 9. Model metrics report.

3.5 Model Evaluation

The model was evaluated for accuracy and loss using the test set. A metrics report and confusion matrix were used to analyze performance by class and identify areas for improvement. The metrics report included precision, recall, and f1-score, the most commonly used metrics to evaluate deep learning models. In addition, the confusion matrix allows visualization of model performance, making it possible to identify which images were classified correctly and which were misclassified. This complements the metrics report.

4 Results

The results of the present study show that the convolutional neural network model has good accuracy in classifying magnetic resonance images for brain tumor detection. The most important results are listed below.

4.1 Dataset Dimensions

The model was trained using an augmented set of 6720 images, validated with 120 original images, and tested with another 120 original images. This data split allowed for correct training and fair evaluation of the model, despite the limited amount of data in the original set.

4.2 Training Evaluation

During the training of the model, significant progress was observed in the ability to correctly classify the images into the four defined categories. At the end of the 20 training epochs, a training accuracy of 95.85% and a loss of 0.1162 was achieved, as can be seen in Fig. 7. In contrast, the validation accuracy was 80.83% with a loss of 0.7878. These metrics indicate a possible slight overfitting of the model to the training set, possibly due to the limited amount of validation data combined with the data augmentation procedure during network training.

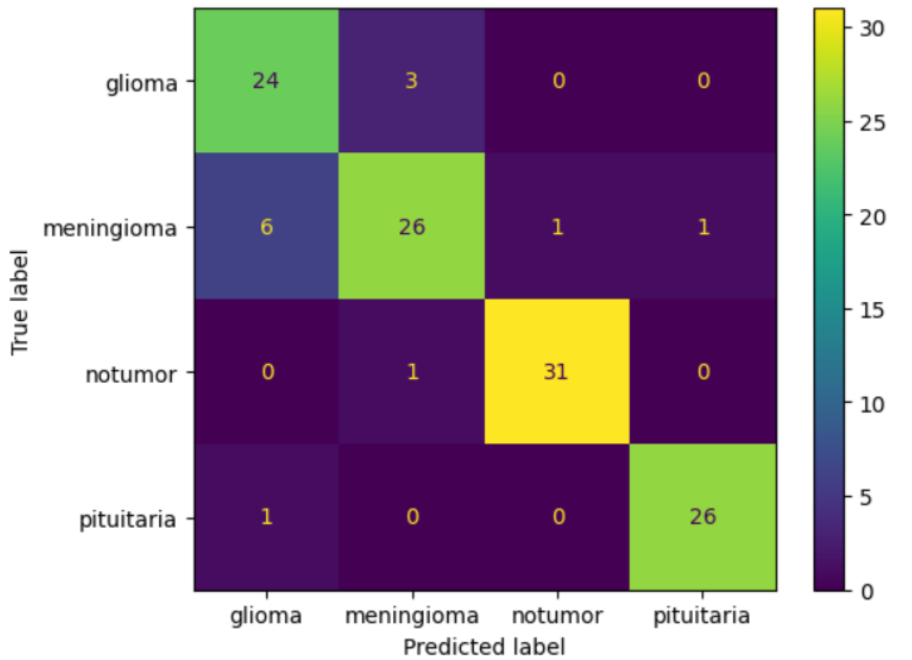


Fig. 10. Confusion matrix for the model.

4.3 Test Evaluation

Fig. 8 shows that evaluation of the model on the test set yielded an accuracy of 89.17% and a loss of 0.5614, confirming the model's ability to generalize to novel images not seen during training. These results are particularly encouraging as efficient classification of brain tumors from magnetic resonance images is challenging due to the variability in tumor appearance.

4.4 Metrics Report

The analysis of the classification metrics shows a high precision, sensitivity (recall), and F1 score in all categories. As shown in Fig. 9, the No Tumor category performed particularly well with a recall of 0.97 and an F1 score of 0.97 (in the report the categories were labeled Glioma $= 0$, Meningioma $= 1$, No Tumor $= 2$, and Pituitary $= 3$), indicating a high true-positive rate and a good balance between precision and sensitivity. The other categories showed comparable results with values between 0.81 and 0.83 for the F1 score for glioma and meningioma and 0.96 for pituitary.

4.5 Confusion Matrix

As can be seen from the obtained confusion matrix, Fig. 10, the model showed a high ability to identify images without tumors, with remarkable accuracy in this category, being able to correctly identify 31 out of 32 cases. However, there were some

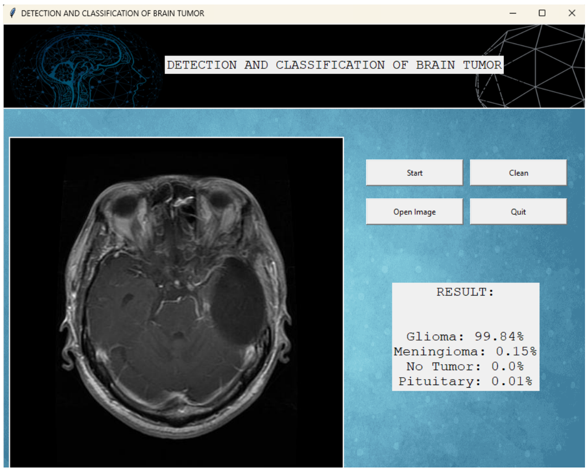


Fig. 11. Prediction for a Glioma MRI image from the test set.

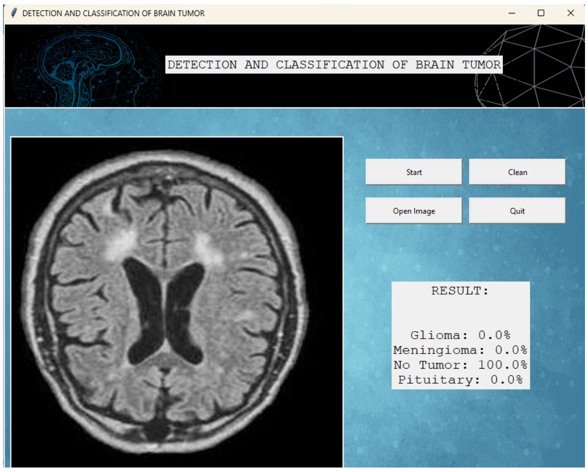


Fig. 12. Prediction for a no tumor MRI image from the test set.

difficulties in accurate classification between tumor types, particularly between glioma and meningioma, where some cases of misclassification were observed.

4.6 Implementation in an Executable Program Using a Graphical Python Interface

The model was implemented in a graphical Python interface by creating an executable file. This was done to allow any user to interact with and use the model without the need to have Python installed on their system and without the need to have technical knowledge of how to use the model. The graphical interface created is quite intuitive, so it can be used by users who are not at all familiar with managing deep learning prediction models. It is based on the Python library tkinter and allows loading MRI

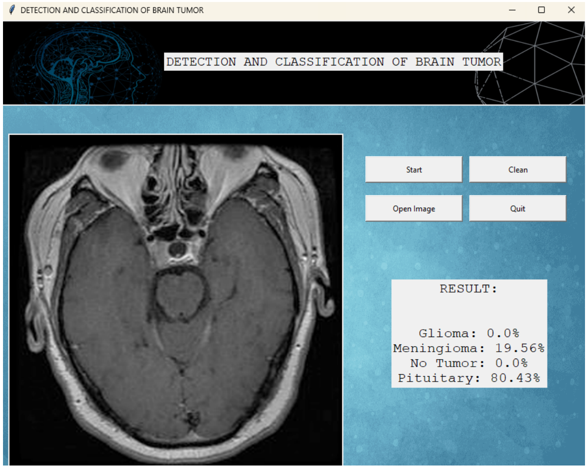


Fig. 13. Prediction for a Pituitary MRI image from [5].

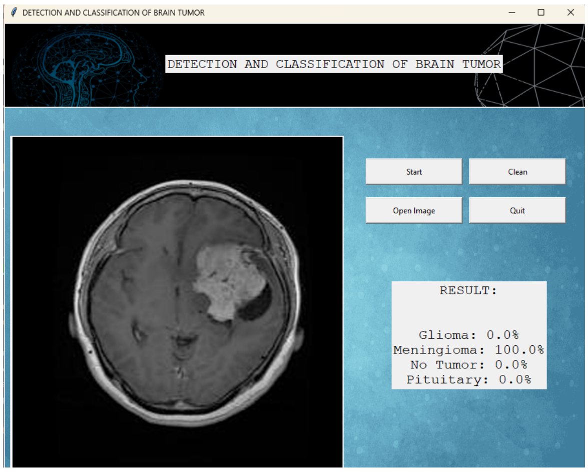


Fig. 14. Prediction for a Meningioma MRI image from [5].

images for later analysis (via the "Open image" button). As a result (after processing and evaluation by the model, if you click on the "Start" button") the loaded image and the membership probabilities predicted by the model for each of the 4 possible classes are displayed. There is also a button to "Clear" the interface, which deletes the results of the last analysis and leaves the interface clean and ready to load a new image, and the "Exit" button, which allows you to close the program at any time. Below is an example of what the graphical user interface looks like after analyzing brain tumor images in the implementation. Figs. 11-12 show the results of the classification of two images from the test set, while Figs. 13-14 are predictions of MRIs from a different dataset [5], showing successful results as well.

5 Conclusions

This study has demonstrated that the use of deep learning techniques, specifically Convolutional Neural Networks (CNNs), together with data augmentation methods, can be of great utility for the detection and classification of brain tumors in Magnetic Resonance Images. The network architecture designed for this research has proven to be effective in classifying gliomas, meningiomas, pituitary tumors, and non-tumors, achieving an overall test set accuracy of 89.17%.

The results obtained suggest several important conclusions. First of all, high performance in the detection of non-tumors (healthy patients) was seen: the model demonstrated an outstanding ability to identify non-tumor images, which is key to preventing false positive diagnoses in clinical settings. Additionally, challenges in tumor type classification were observed: although the model has shown high overall accuracy, it still faces challenges in differentiating between certain tumor types, such as glioma and meningioma. This could indicate a need for further adjustments to the model architecture or training approach to improve specificity or a need for a larger amount of training data and more data for each class.

Likewise, data augmentation proved to be essential in improving the model's ability to generalize to new images, as initially the data set available was notably limited, and having trained with this set would likely have led to inferior results due to the restricted number of examples in each class. This approach could be further explored to include other augmentation techniques that could help improve the distinction between similar classes. On the other hand, the difference between training and validation accuracy suggests overfitting, a more or less expected result due to the limited validation set. Strategies such as regularization, dropout, adding callbacks to the training process, or increasing the validation set could be investigated to mitigate this effect.

Finally, the fact of having created a Python graphical interface compiled as an executable file contributed greatly to making the use of the model easier and more intuitive, and added value to the final work. In this way, the use of this type of predictive model is brought closer to the common user, who does not necessarily need to know the strategies that must be followed normally to be able to use such models correctly, for instance, when following a Jupyter notebook or executing a Python script.

Future work could explore the optimization of the network architecture, fine-tuning of hyperparameters from larger pre-trained networks, and expansion of the data set to improve both the training process and the validation and evaluation process (possibly by collaborating with medical institutions to acquire more images and getting in touch with specialists from these institutions to start developing a way to utilize the graphical interface to assist in making appropriate diagnoses). In turn, the incorporation of other data modalities, such as clinical patient data, could help improve the accuracy and robustness of the model.

Acknowledgments. The authors would like to thank the Mexican Council of Science and Technology for the scholarship they received, and Dr. Sebastián Salazar-Colores for encouraging them to write it and submit it to contest. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Al-Zoghby, A. M., Al-Awadly, E. M. K., Moawad, A., Yehia, N., Ebada, A. I.: Dual deep cnn for tumor brain classification. Diagnostics, vol. 13, no. 12, pp. 2050 (2023) doi: 10.3390/diagnostics13122050
2. Amin, J., Sharif, M., Gul, N., Raza, M., Anjum, M. A., Nisar, M. W., Bukhari, S. A. C.: Brain tumor detection by using stacked autoencoders in deep learning. Journal of Medical Systems, vol. 44, no. 32 (2019) doi: 10.1007/s10916-019-1483-2
3. Bakas, S., Reyes, M., Jakab, A., Bauer, S., Rempfler, M., Crimi, A., Shinohara, R. T., Berger, C., Ha, S. M., Rozycki, M., Prastawa, M., Alberts, E., Lipková, J., Freymann, J. B., Kirby, J. S., Bilello, M., Fathallah-Shaykh, H. M., Wiest, R., Kirschke, J., Wiestler, B., et. al.: Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the BRATS challenge (2018) doi: 10.48550/arXiv.1811.02629
4. Bhanothu, Y., Kamalakannan, A., Rajamanickam, G.: Detection and classification of brain tumor in MRI images using deep convolutional network. In: Proceedings of the 6th International Conference on Advanced Computing and Communication Systems, pp. 248–252 (2020) doi: 10.1109/icaccs48705.2020.9074375
5. Cheng, J.: Brain tumor dataset. Figshare (2024) doi: 10.6084/m9.figshare.1512427.v57
6. Das, S., Riaz-Rahman-Aranya, O. F. M., Labiba, N. N.: Brain tumor classification using convolutional neural network. In: Proceedings of the 1st International Conference on Advances in Science, Engineering and Robotics Technology , pp. 1–5 (2019) doi:10.1109/icasert.2019.8934603
7. Gupta, S.: Brain MRI Scans for brain tumor classification (2023) www.kaggle.com/datasets/shreyag1103/brain-mri-scans-for-brain-tumor-classification/data
8. Han, C., Rundo, L., Araki, R., Furukawa, Y., Mauri, G., Nakayama, H., Hayashi, H.: Infinite brain mr images: PGGAN-based data augmentation for tumor detection. Neural Approaches to Dynamics of Signal Exchanges. Smart Innovation, Systems and Technologies, vol. 151, pp. 291–303 (2019) doi: 10.1007/978-981-13-8950-4_27
9. Menze, B. H., Jakab, A., Bauer, S., Kalpathy-Cramer, J., Farahani, K., Kirby, J., Burren, Y., Porz, N., Slotboom, J., Wiest, R., Lanczi, L., Gerstner, E., Weber, M., Arbel, T., Avants, B. B., Ayache, N., Buendia, P., Collins, D. L., Cordier, N., Corso, J. J., et. al.: The multimodal brain tumor image segmentation benchmark (brats). IEEE Transactions on Medical Imaging, vol. 34, no. 10, pp. 1993–2024 (2015) doi: 10.1109/tmi.2014.2377694
10. Mohsen, H., El-Dahshan, E. A., El-Horbaty, E. M., Salem, A. M.: Classification using deep learning neural networks for brain tumors. Future Computing and Informatics Journal, vol. 3, no. 1, pp. 68–71 (2018) doi: 10.1016/j.fcij.2017.12.001
11. Nazir, M., Shakil, S., Khurshid, K.: Role of deep learning in brain tumor detection and classification (2015 to 2020): A review. Computerized Medical Imaging and Graphics, vol. 91, pp. 101940 (2021) doi: 10.1016/j.compmedimag.2021.101940
12. Özkaraca, O., Bağrıaçık, O. İ., Gürüler, H., Khan, F., Hussain, J., Khan, J., Laila, U. E.: Multiple brain tumor classification with dense CNN architecture using brain MRI images. Life, vol. 13, no. 2, pp. 349 (2023) doi: 10.3390/life13020349
13. Rai, H. M., Chatterjee, K.: Detection of brain abnormality by a novel Lu-Net deep neural CNN model from mr images. Machine Learning with Applications, vol. 2, pp. 100004 (2020) doi: 10.1016/j.mlwa.2020.100004
14. Srinivasan, S., Francis, D., Mathivanan, S. K., Rajadurai, H., Shivahare, B. D., Shah, M. A.: A hybrid deep CNN model for brain tumor image multi-classification. BMC Medical Imaging, vol. 24, no. 1 (2024) doi: 10.1186/s12880-024-01195-7

15. Ucuzal, H., YASAR, S., Colak, C.: Classification of brain tumor types by deep learning with convolutional neural network on magnetic resonance images using a developed web-based interface. In: Proceedings of the 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies, pp. 1–5 (2019) doi: 10.1109/ismsit.2019.8932761
16. Vankdothu, R., Hameed, M. A., Fatima, H.: A brain tumor identification and classification using deep learning based on cnn-lstm method. Computers and Electrical Engineering, vol. 101, pp. 107960 (2022) doi: 10.1016/j.compeleceng.2022.107960

ISSN 1870-4069

From Simple Detection to Quality-aware Prediction: Exploring Argument Complexity with Machine Learning

Anvi Alex-Eponon, Muhammad Tayyab-Zamir, Lemlem Kawo-Eyob, Luis Israel Ramos-Perez, Ildar Bartyrshin, Grigori Sidorov, Olga Koleniskova, Francisco Hiram Calvo-Castro, Alexander Gelbukh

Instituto Politécnico Nacional,
Centro de Investigación en Computación, Mexico City, Mexico

{epononanvi, tayyab.awan8001, lemlemeyob19, kolesolga}@gmail.com,
{lramos2020, sidorov, batyr1, hcalvo, gelbukh}@cic.ipn.mx

Abstract. Argument mining is a critical area within artificial intelligence with significant implications for the future of machine learning models. It is widely believed that advances in argument mining will enhance the ability of models to construct more effective arguments in diverse contexts, including educational and political settings. However, existing research predominantly focuses on identifying argument structures without sufficiently considering the nuanced quality dimensions inherent within them. This study addresses this gap by conducting several experiments. Firstly, it evaluates the performance of traditional machine learning models in detecting arguments. Subsequently, the research investigates how selected quality dimensions impact the performances of argument prediction. The methodology leverages BM25 features with a Random Forest model, achieving notable results with an F1-score of 0.88 and a Spearman's correlation coefficient of 0.73. These outcomes surpass those of previous models such as IBM's 2019 Arg-ranker and base-Arg-ranker, which utilized Bert embeddings and achieved Spearman's scores of 0.41 and 0.42 respectively.

Keywords: Argument mining, machine learning, deep learning, argument quality assessment.

1 Introduction

An argument in general is a combination of sentences or paragraphs that tries to convey a reason or many reasons to specific conclusions. By so doing, an argument can be seen as a 'system of reasoning' for providing or arriving at a particular state, being logical, dialectical, rhetorical, true, false, good, or bad.

In Linguistics and computer science, Argument mining becomes crucial for machines to understand the real reasoning behind the human language. The works on arguments theoretically and philosophically speaking have been significantly influenced by Aristotle [3]. In almost every setting in the society, arguments are used. In political debates, online discussions via social media, educational settings online product reviews, or even in written books either scientific or fictional, arguments are presented. So Argument mining becomes more and more important in the field of artificial intelligence specifically in the area of Natural language understanding as it helps spread light on how humans reason to communicate effectively through language.

Thus several studies have been done in artificial intelligence concerning this task. It is worth noting that even though many research studies have been conducted on argument mining, which is the field of artificial intelligence aiming at automatically detecting and extracting argumentative structures and their relations from text, several challenges still need to be uncovered. One of them is the identification and the impact of dimension qualities of arguments in various settings or domains of applications.

A good explanation is that the Logic dimension is the one which is the most used in scientific settings such as mathematics where coherence is more important than finding truth while the Dialectical dimension which appears mostly in social avenues deals with finding the truth, what should be acceptable, agreed on or not, etc. It appears then that studying the argument dimensions and their qualities through Natural Language Processing (NLP) techniques will enhance the understanding of argumentation in general for machines but also for specific cases, which could impact positively the way machines model the reasoning behind the human language. The current study tries to bring answers to these two specific questions:

– Can traditional machine learning (ML) models identify arguments using state-of-the-art Natural Language Processing techniques such as Best Match 25 (BM25)?

– Given the specific quality dimensions from the dataset, can they enhance the performance of Deep Learning models in discriminating arguments compared to traditional ML? (This question addresses both the evaluation of the dataset and the identification of the structural complexity of arguments).

2 Literature Review

Argument mining can be defined as the action of identifying and extracting the structure of an argument in natural language and the inferences and reasons behind it. This way, knowing argumentative structures, an understanding is built not only from where people stand but also the reasons they have for doing so. This is useful in several contexts, ranging from the prediction of financial markets to public relations [8], also it has been applied in political debates, online discussions, and customer reviews [4, 15]. Argument mining has been a major topic in the Natural language processing literature [23]. In several domains, Argument mining has been explored. For example, argumentation in learning has been found to have the effect of enhancing argumentation skills among students, and computational models of argumentation have been synthesized to enhance this process [9]. One aspect of argument mining is the use and identification of dimensions present in arguments that make them strong or not for a particular purpose. A lot of dimension qualities have been studied and designed, but one of the most

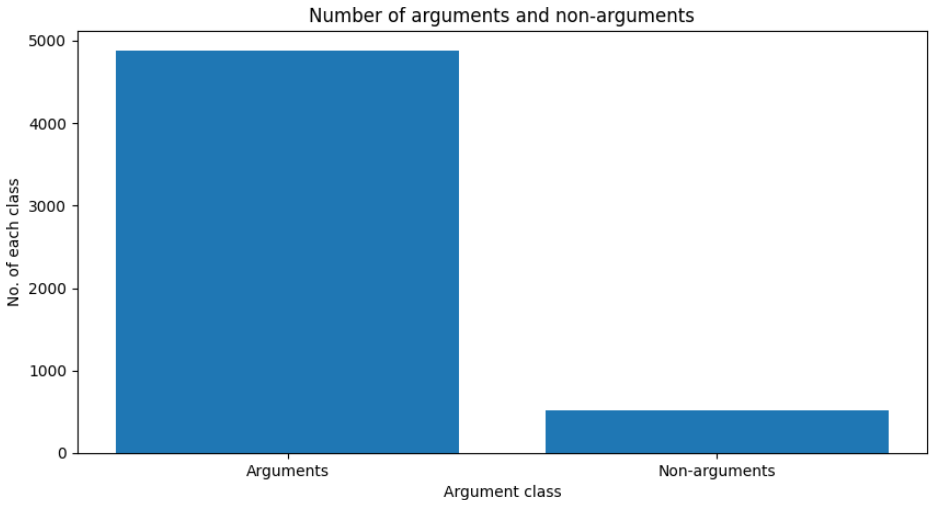


Fig. 1. Argument vs non-argument proportions.

recurrent in the literature are rhetorical, logical, and dialectical quality [13]. Analyzing arguments is considered essential for understanding public discourse and enhancing critical thinking skills [26].

Different techniques and models are employed for argument mining and argument quality analysis. In fact, in [13], twelve qualities have been identified related to the Logic dimensions of arguments but only three relate more with Logic which are Cogency, Fallaciousness, and Strength [11, 6, 7], only one to Rhetoric, which is Effectiveness [22] and finally three to Dialectic, which are Convincingness, Reasonableness, and Global sufficiency [1, 2, 5]. A total of 25 qualities have been identified in this research related to only three dimensions.

For instance, in the research [14], authors discuss three methods for extracting the argumentative structure from a piece of natural language text. The first method uses discourse indicators to determine argumentative relationships between nearby propositions in a text. The second method uses topic changes to classify argument components and identify their relationships with supervised machine learning. The last method is concerned with the capability of combining all these individual techniques to enhance argument structure identification.

In this paper, the authors report the first complete work on computational argumentation quality in natural language. They summarize the broad range of existing theories and approaches for considering the logical, rhetorical, and dialectical quality aspects, out of which taxonomy is developed systematically. It also contributes 320 argumentation cases that have been annotated for all of the 15 dimensions, for instance, Cogency, Local relevance, Local sufficiency, Well Formedness, Effectiveness, Arrangement, appropriateness of style, Convincingness, Global acceptability, Reasonableness.

The research findings provide the basis for comparison for research on computational approaches to argument quality assessment [25]. Another study explores current NLP feedback systems by categorizing each into four important dimensions

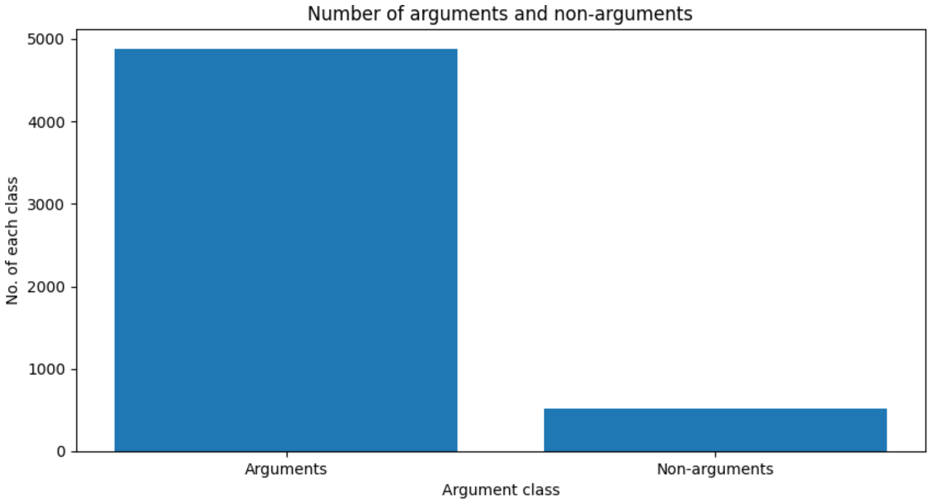


Fig. 2. Proportion of dimensions.

of feedback: The four major areas for improvement are richness, visualization, interactivity, and Personalization. Each of the dimensions is also reviewed in terms of its drawbacks, and recommendations for feeding and explanation are given with the aim of developing users' critical thinking capabilities [9].

In the work of [17], argument relevance is analyzed based on user perception. This paper attempts to make the first study on this dimension to establish the foundation for the future advancement of the technology the authors reviewed over 300,000 arguments using four retrieval models across forty topics on twenty controversial issues, considering both biased and neutral perspectives. However, few works in NLP have been done on the importance of dimension qualities cited earlier in the prediction of arguments on different settings or domains.

Most of them focused on the overall argument detection itself, or its structures. In the study [21], the authors state that BERT outperforms most baselines for modeling causational hierarchies in typical argument structures within online discourse. This model generates embeddings, which are then processed through a transformer encoder layer to identify edges between them. Another study proposes the creation of a written corpus for argumentative reasoning, analyzed with advanced argumentation techniques, and marked up using an open, reusable language.

It highlights how this resource can be used in linguistic, computational, and philosophical research and also discusses its role in initiating a program for automatic detection of argumentative structure [18]. The advancement of artificial intelligence also benefits argument mining with the use of deep learning models and large language Models (LLMs). The work [16] involves using LLMs as argument quality annotators and evaluating the agreement between LLMs, human experts, and novices based on argument quality dimensions.

LLMs show moderate agreement with experts and improve inter-annotator consistency, proving valuable for automated argument quality assessment of large datasets. [24] carried out a review of the literature on argument quality and

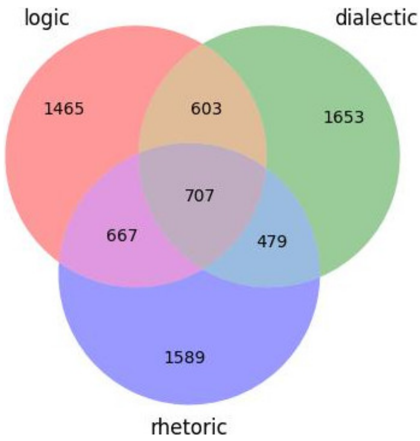


Fig. 3. Overlap between dimensions.

Table 1. Best hyperparameters for different models.

Models	Best Hyperparameters
Logistic Regression	Only TF-IDF (1-3 grams)
Naive Bayes	Only TF-IDF (1-3 grams)
SVM	Classifier C: 1, Kernel: Linear
Random Forest	n_estimators: 200

suggests using instruction-following LLMs for assessment, stresses systematic training with argumentation theories and examples, and discusses practical implementation, including benefits and moral considerations.

In [12] the researchers describe the first dialogue conference competition for recognizing argumentation analysis of Russian language texts. It included a stance detection task and argument classification with a dataset of 9,550 comments gathered from various social media platforms regarding COVID-19 topics. The presented NLI-BERT-TargetMask obtained F1-scores of 0. 6968 and 0.7404 for stance detection and argument classification in particular.

The research [10] proposes (What Is Being Argued?) WIBA is a new framework to address what is being argued in a range of settings. Their approach identifies the existence of the argument, its topic, and its stance, using the fine-tuning of LLMs. They get an F1 of between 79%-86%, the method of identifying topics gets an average of 71% similarity, and the Stance Classification method gets 71%-78% F1. The authors concluded that WIBA facilitates analysis of the arguments in large contexts and across the domains of linguistics, communication, social, and computer sciences.

Finally, in the work of [22], an end-to-end approach is proposed for jointly predicting all predicates, argument spans, and the relations between them. The model independently determines what relationship, if any, exists between every possible word-span pair and learns contextualized span representations that offer rich, shared input features for each decision.

Table 2. Experimental setup for different models.

Models	**Epochs**	**k-folds**	**Batch Size**	**Learning Rate**
BiLSTM 1	10	5	64	0.001
BiLSTM 2	10	0	64	0.001
CNN Numeric	10	0	64	0.001
CNN textual	10	5	64	0.001

In paper [23], the data used are the discourse of students and annotations that were obtained from the Kaggle platform. They use DeBERTa for predicting effective arguments. The lowest of the metric is achieved by the Deberta-large which owns 0.619 among these models, which is 0.007, 0.114, and 0.030 lower than BERT, and RoBERTa respectively. As observed, the literature on argument mining has less focus on the importance of dimension qualities of arguments and how they impact the strengths of arguments in different settings.

The objectives of such previous research were to theoretically identify the quality dimensions without evaluating the impact of their presence in identifying arguments. The current study aims to introduce a series of studies that aim to present with NLP techniques the performances on the identification of argument dimension qualities and their impact on arguments. This ablation approach mainly missing in previous studies focusing on identifying dimension qualities brings insights into the importance of the quality dimensions features in argument mining.

3 Methodology

3.1 Assumptions and Task Objectives

The methodology designed for the current study aligns with the objectives and assumptions made. To conduct the experiments, some assumptions were made regarding the nature of an argument and how it could be identified. Throughout the experiment, we assumed that a good argument has a well-defined structure (either inductive or deductive related in part, to a syntactic nature) but also aligns with the understanding of the target audience (contextual nature).

Not only that, we defined a “good” argument as a sentence or a paragraph that contains a conclusion or an opinion related to illustrations (personal or general) and is possibly supported by regulation facts. This definition does not attribute any Truthfulness to arguments, in other sense that in our study we don’t assess an argument as being “good” because it is accepted as “true” but rather if the statement given earlier has a coherence between the conclusion, illustrations, and regulations. This assumption is supported by the fact that any argument can be “strong” or “good” without being necessarily “true” as the notion of “Truth” can be ambiguous. An example can be observed in Legal statements such as:

“The defendant should be acquitted because there is no conclusive evidence linking them to the crime”. – Refers to the legal term Acquittal (US Legal Terms Glossary).

The argument stated is strong with respect to the legal context, and guilt in this context, must be proven beyond reasonable doubt. However, The truth of the actual involvement of the defendant in the crime remains uncertain; the argument is based on the current legal standard rather than an objective truth about the guilt of the defendant or innocence where this standard can change from one culture to another. Then, the study is divided into two tasks. The first one is to implement and evaluate the performances of Machine learning models in predicting arguments on several topics either in political debates or online review quality assessments with a leading question:

Can Machine learning models identify easily and effectively arguments from statements that are not (considered from the annotators point of view) using Natural Language Processing techniques? The second task has the purpose of evaluating deep learning models limited to Bilstm and CNN models but this time with combinations of features related to dimension qualities of arguments to analyze if the selected qualities in the datasets help identify arguments or not or if their presence or not impact the strength of the argument.

3.2 Dataset and Exploratory Data Analysis (EDA)

Dataset. To conduct the overall study, two datasets were used. One from the IBM Debater datasets [23] was made specifically for Argument quality. This dataset contains more than 34,000 samples of arguments focusing on identifying the better ones. And the second dataset is from Gaqcorpus [13]. In the current experiment, we used a portion of the IBM Debater datasets containing a bit more than 23,000 samples. The dataset was presented during the EMNLP conference led in 2019 which contains 5 times more samples than the UKPRank dataset [23]. The whole Gaqcorpus dataset containing 6,424 samples was used in the study. This dataset [13] fills the gap by bringing a large-scale (more than 5000 arguments) English multi-domain corpus (Debate forums, Community Question Answering, Reviews) annotated with a theory-based Argument-quality score.

EDA (GaqCorpus). The GaqCorpus dataset introduces a textual English corpus of arguments in several domains such as debate forums, review forums, or community Q&A forums. The dataset is comprised of 6,424 premises and conclusions associated with quality features such as degree of:

- Logic,
- Dialectic,
- Rethoric,
- Relevance.

The dataset contains 4,873 considered arguments and only 513 considered nonarguments. Which creates a fair imbalance dataset for the binary prediction of arguments. Out of the arguments, 3,442 relevant arguments have been identified (threshold put on relevance $\geq$ 3 out of 4). The relevant arguments are dispatched as follows (Fig.2 and Fig.1):

- 3,388 logical arguments;

Table 3. Model Performance Comparison on 23,000 samples.

Model	F1	Spearmanr
Logistic Reg	0.7741	0.5452
Naive Bayes	0.7775	0.5474
SVM	0.7749	0.5464
Random Forest	**0.7706**	**0.5391**

– 3,122 dialectical arguments;

– 3,212 rhetorical arguments.

3.3 Task 1: Binary Argument Prediction from Premises Only

Task 1 is about finding if traditional machine learning models can effectively predict arguments given a premise. For this end, we used a portion of the IBM Debater dataset which contains +23,000 samples of pairs of arguments which is approximately 67.94% of the full dataset. The final dataset after preprocessing contains two columns. The premise and the label. To handle this experiment, two separate experiments have been done. The first one involves training traditional machine learning models with TF-IDF features, and the second one on BM25 features without making any preprocessing on the textual data with the assumption that the original structure of the text is crucial to identifying arguments.

Feature Extraction Phase. During this phase, only two techniques were selected. The first one commonly used is the Term Frequency Inverse Document Frequency (TF-IDF). Even though in the current datasets, premises are small in size compared to essays or political speeches, applying TF-IDF to the datasets, will help the models identify patterns or important words related to the identification of arguments. The second feature extraction technique used in this part of the experiment is the Best Matching 25 (BM25) which has been proven excellent in the literature as a ranking function [20].

Model Selection and Experimental Phase. For model selections task, four models were used which are:

– Random Forest,

– Logistic Regression,

– Support Vector Machine (SVM),

– Naive Bayes.

Random Forest has been proven efficient in several tasks of classification in Machine learning. Due to its capacity to learn implicit features from different sub-trees and also its robustness to overfitting, Random Forest has been chosen. Due to the binary nature of the task, logistic regression has been chosen. Its efficiency over large datasets and capacity to differentiate between two classes with a sigmoid function make it suitable for our experiments. Support Vector Machine algorithm has been chosen in this task

Table 4. Performance comparison–7,709 samples.

	F1	**Spearmanr**
Logistic Reg	0.5378	0.1135
Naive Bayes	0.5413	0.0412
SVM	0.5501	0.1072
IBM Arg-ranker	–	0.42
Random Forest	**0.8855**	**0.731**

first, for its performance to overfitting like Random Forest, but also for its ability to handle non-linearity over the features that happen on complex textual datasets. Lastly, Naive Bayse has been selected for the experiment due to its ability to handle many features or large vocabulary.

Further we apss to the experimental phase, where we first perform selectoin of Hyperparameters for Feature Extraction. Concerning the feature extractions used, all the models used TF-IDF features on n-grams varying from 1 to 3. Additionally, after experiments, we set the number of estimators for the random forest model at 200 and set the kernel parameter of the SVM model to linear.

Training Phase. The next step is training phase. The models selected were implemented from the scikit-learn library. Most of the parameters have been left by default except the ones mentioned earlier to better measure the performance of the models.

3.4 Task 2: Binary Predictions from Quality Dimensions and Premises

Task 2 has been conducted by using solely the dataset provided by Gaqcorpus [13]. This dataset contains +6,000 samples of arguments scored based on their dimension qualities. This task also tries to understand if deep learning models perform better on argument detection but also if the presence of the selected dimension qualities impacts this detection. To answer these questions, the study has been divided into two experiments too each related to one model, a Bidirectional Long-Short Term-Memory (BiLSTM) model and a Convolutional Neural Network (CNN).

Preprocessing Phase. The preprocessing phase of the experiments made with the BiLSTM models involves a few transformation steps of the texts such as:

– Remove of English stop words;

– Lemmatization of tokens;

– Part of Speech Tagging.

However, only the removal of stopwords has been done on the second BiLSTM model in order to see the impact of syntactic processing on the performance of the models. Concerning the CNN models, the first one has been trained on the quality dimensions of the arguments while the second has been trained solely with the premises to detect whether the qualities give an advantage for the predictions of arguments.

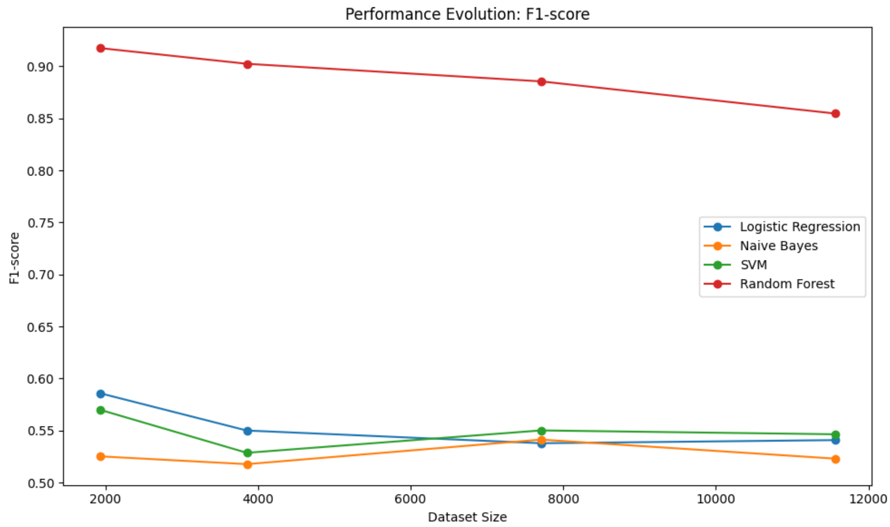


Fig. 4. F1-score evolution. Performance evolution using BM25.

Feature Extraction Phase. We used one deep word embedding model for all the models selected which is SentenceBert [19]. The deep word embedding model used is from the sentenceBert library.

Model Selection and Experimental Phase. For model selection, two models were used which are:

– BiLSTM;

– CNN.

Bidirectional Long-Short Term-Memory(LSTM) is well suited for the second task due first to its capability to handle sequential input data. It has a longer memory dependency compared to the model used in the previous task but also with its ability to have a combination of context from both directions of the sequence, it can capture more detailed features related to context. On the other hand, CNN also proved to be efficient in Hierarchical feature learning where in the context of argument mining this capability is crucial.

Then we pass to the Experimental Phase. First we select Hyperparameters for Feature Extraction.

4 Results

4.1 Task 1 Results

The research has been conducted through several experiments. The first Task, which is comprised of baseline experiments and improved baseline considered four traditional machine learning models as mentioned in the methodology. The baseline experiments have been conducted on the 23,000 samples while the baseline improved using BM25

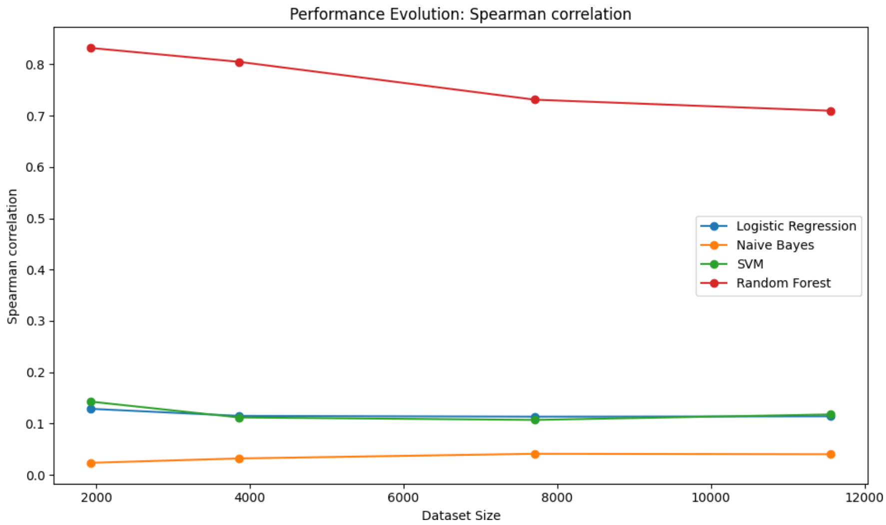


Fig. 5. Spearman correlation evolution. performance evolution using BM25.

Table 5. Model performance comparison of deep learning models.

Model	F1	Spearmanr
BILSTM_1	**0.8812**	**0.7876**
BILSTM_2	0.8812	0.7876
CNN_numeric	0.8812	0.7816
CNN_textual	**0.8815**	**0.7881**

Table 6. Ablation experiments on dimension qualities.

Feature Excluded	Accuracy	F1-score
cogency_(logic)	0.7875	0.8811
effectiveness_(rhetoric)	**0.7876**	**0.8812**
reasonableness_(dialectic)	0.7876	0.8812

used half of the dataset, which is approximately 11,000 samples. Below are the results concerning the baseline experiments. The best results of the second part of Task 1 also called improved baseline are recorded in the table below. The figure below shows the evolution of the models according to the increase in the sample size.

4.2 Task 2 Results

Task 2 considered all the 23,000 samples of arguments to train two Deep learning models (CNN, BiLSTM). The tables 5 and 6 displays the results of the models.

A last ablation experiment has been conducted on the BiLSTM model in order to uncover the real impacts of the dimensions selected in the datasets. Below are presented the results at the last fold of the training using either one of the dimensions for predictions.

5 Discussions

As mentioned earlier in the current study, the objectives are to discover if traditional machine learning models could effectively predict arguments given premises and finally evaluate deep learning models on predicting arguments based on several quality dimensions. In the first task, specifically in the baseline experiments, all traditional models perform just above luck concerning the Spearmanr score and at more than 0.70 f1 scores, beating the IBM Arg-ranker-base model with a minimum Spearmanr score of 0.53. These results, at the initial step of the experiment, might be due to the size of the dataset, where 23,000 samples were used compared to approximately 6,000 samples in the case of the IBM Arg-ranker-base model.

However, this initial baseline does not make use of any contextual embedding such as Bert, rather relies on bag-of-word techniques specifically TF-IDF. The second experiment in task 1 which is the baseline improved, makes use of the BM25 ranking function in order to predict arguments. Due, to the heavy computational resources needed, only half of the dataset was used, thus approximately 11,000 samples. Except for the Random Forest model, all the models perform at luck gradually decreasing in performance with the increase of the samples.

This highlights the complexity of the features embedded in the premises since all the models were trained on their best parameters. The performance that can be compared with the IBM Arg-Ranker obtained at samples equal to 7,709 where the Random Forest performs at 0.88 of f1-score with 0.73 for the Spearmanr score, which performs better than the IBM Arg-Ranker-based but also the IBM Arg-ranker which were trained on vanilla Bert and finetuned Bert embeddings.

This performance of the Random Forest model might be due to its capability to detect and associate specific structures to arguments from the subtrees. On the other hand, the experiments on the deep learning models show that deep learning models are much more stable in predicting arguments either from premises only or with quality dimensions. This is observed by the constant f1 score turning around 88% at each epoch and fold experiment in the ablation experiment. However, from those same experiments, the presence of the quality dimensions does not influence significantly the identification of arguments. This could be due to the fact that most of the dimensions scores overlap as seen in the EDA study, which might indicate a limit in the annotation process.

6 Conclusions

The task of argument mining even though presenting several interests in the literature remains a task with several challenges in NLP. Constructing models that leverage understanding of the human language to generate correct arguments can influence several sectors of society. In the present work, experiments showed that the identification of arguments can be effectively done by traditional models with the correct feature extractions such as BM25 ranking functions. However, if deep learning models such as BiLSTM and CNN can be more stable and capture more complex hidden features the question of which quality dimensions impact this prediction is still unanswered. The development of dimensions and quality dimensions in argument has been a serious topic since the Ancient Greeks. Finding an automatic approach to learning the correct argument construction will push forward the performances of future models. Finally, in our study, Random Forest performs the best at 0.88 for the f1-score and 0.73 for the Spearmanr score with approximately 7,000 samples of arguments which creates a new baseline surpassing the baseline proposed by IBM Args-ranker which lies at 0.42 for the Spearmanr score.

7 Limits of the Study and Future Work

The current study presents itself as an introduction to a series of experiments to be conducted in argument mining specifically in the modeling of dimension qualities. Thus it has been done with a lot of limitations. The first and main one is the experiments focused on the influence of dimension qualities on the predictions of arguments regardless of the specific domain in which the arguments have been constructed. Given that different fields have their own rules and methods for making arguments, the current approach struggles to distinguish between these different characteristics specific to each field. For example, the way arguments are constructed in Political debate and online reviews are not the same. From this perspective, how does understanding the unique qualities of each field impact not only the prediction of the argument but also the understanding of the domain? Alternatively, how can knowledge of the domain help in predicting the quality dimensions present in a given argument? Those are questions the current research is not answering. The second limit lies in the Gaqcorpus dataset itself. The annotating procedure ended up with qualities that significantly overlap (Fig.3). This brings a lot of ambiguities in differentiating the three dimension qualities.

A better approach to annotating such information might need to be addressed. Also, using only two datasets, may not fully represent the diversity of argument structures and qualities across different domains. Thus a construction of a diverse argument dataset covering different languages can be an adequate future avenue. Finally, the development of NLP techniques such as tokens or sets of n-tokens specifically targeting dimension qualities in an argument could present several advantages in effectively detecting the dimension qualities.

Acknowledgments. The work was done with partial support from the Mexican Government through the grant A1- S-47854 of CONACYT, Mexico, grants 20241816, 20241819, and 20240951 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

1. Aikin, S. F.: Perelmanian universal audience and the epistemic aspirations of argument. Philosophy and Rhetoric, vol. 41, no. 3, pp. 238–259 (2008)
2. Andone, C.: A systematic theory of argumentation. The pragma-dialectical approach. Journal of Pragmatics, vol. 37, no. 4, pp. 577–583 (2005) doi: 10.1016/j.pragma.2004.07.003
3. Bench-Capon, T., Dunne, P. E.: Argumentation in artificial intelligence. Artificial Intelligence, vol. 171, no. 10–15, pp. 619–641 (2007) doi: 10.1016/j.artint.2007.05.001
4. Chakrabarty, T., Hidey, C., Muresan, S., McKeown, K., Hwang, A.: Ampersand: argument mining for persuasive online discussions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 2933–2943 (2019) doi: 10.18653/v1/d19-1291
5. Cohen, J.: Defining identification: A theoretical look at the identification of audiences with media characters. Mass Communication and Society, vol. 4, no. 3, pp. 245–264 (2001) doi: 10.1207/s15327825mcs0403_01
6. Damer, T. E.: Attacking faulty reasoning: A practical guide to fallacy-free arguments (1980) ci.nii.ac.jp/ncid/BA80518138
7. Govier, T.: A practical study of argument (1985) ci.nii.ac.jp/ncid/BB16391451?l=en
8. Guerraoui, C., Reisert, P., Inoue, N., Mim, F. S., Singh, K., Choi, J., Robbani, I., Naito, S., Wang, W., Inui, K.: Teach me how to argue: A survey on NLP feedback systems in argumentation. In: Proceedings of the 10th Workshop on Argument Mining, pp. 19–34 (2023) doi: 10.18653/v1/2023.argmining-1.3
9. He, L., Lee, K., Levy, O., Zettlemoyer, L.: Jointly predicting predicates and arguments in neural semantic role labeling. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 364–369 (2018) doi: 10.18653/v1/p18-2058
10. Irani, A., Park, J. Y., Esterling, K., Faloutsos, M.: WIBA: What is being argued? a comprehensive approach to argument mining (2024) doi: 10.48550/ARXIV.2405.00828
11. Johnson, R. H., Blair, J. A.: Logical self-defense (1983) ci.nii.ac.jp/ncid/BA26931398
12. Kotelnikov, E., Loukachevitch, N., Nikishina, I., Panchenko, A.: Ruarg-2022: Argument mining evaluation. In: Proceedings of the International Conference "Dialogue", pp. 333–348 (2022) doi: 10.28995/2075-7182-2022-21-333-348
13. Lauscher, A., Ng, L., Napoles, C., Tetreault, J.: Rhetoric, logic, and dialectic: advancing theory-based argument quality assessment in natural language processing. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 4563–4574 (2020) doi: 10.18653/v1/2020.coling-main.402
14. Lawrence, J., Reed, C.: Combining argument mining techniques. In: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 127–136 (2015) doi: 10.3115/v1/w15-0516
15. McCloskey, B. J., LaCasse, P. M., Cox, B. A.: Natural language processing analysis of online reviews for small business: Extracting insight from small corpora. Annals of Operations Research, vol. 341, no. 1, pp. 295–312 (2024) doi: 10.1007/s10479-023-05816-2
16. Mirzakhmedova, N., Gohsen, M., Chang, C. H., Stein, B.: Are large language models reliable argument quality annotators? Robust Argumentation Machines, Lecture Notes in Computer Science, vol. 14638, pp. 129–146 (2024) doi: 10.1007/978-3-031-63536-6_8
17. Potthast, M., Gienapp, L., Euchner, F., Heilenkötter, N., Weidmann, N., Wachsmuth, H., Stein, B., Hagen, M.: Argument search: assessing argument relevance. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1117–1120 (2019) doi: 10.1145/3331184.3331327
18. Reed, C., Palau, R. M., Rowe, G., Moens, M.: Language resources for studying argument. In: Proceedings of the 6th International Conference on Language Resources and Evaluation (2008)

19. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 4512–4525 (2020) doi: 10.18653/v1/2020.emnlp-main.365
20. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at trec-3. In: Overview of the Third Text REtrieval Conference, pp. 109–126 (1994)
21. Srivastava, P., Bhatnagar, P., Goel, A.: Argument mining using bert and self-attention based embeddings. In: Proceedings of the 4th International Conference on Advances in Computing, Communication Control and Networking, pp. 1536–1540 (2022) doi: 10.1109/icac3n56670.2022.10074559
22. Tang, T.: Predicting effective arguments with a natural language processing model. In: Proceedings of the 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology, pp. 436–439 (2022) doi: 10.1109/cei57409.2022.9950105
23. Toledo, A., Gretz, S., Cohen-Karlik, E., Friedman, R., Venezian, E., Lahav, D., Jacovi, M., Aharonov, R., Slonim, N.: Automatic argument quality assessment - New datasets and methods. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pp. 5625–5635 (2019) doi: 10.18653/v1/d19-1564
24. Wachsmuth, H., Lapesa, G., Cabrio, E., Lauscher, A., Park, J., Vecchi, E. M., Villata, S., Ziegenbein, T.: Argument quality assessment in the age of instruction-following large language models. In: Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation, pp. 1519–1538 (2024)
25. Wachsmuth, H., Naderi, N., Hou, Y., Bilu, Y., Prabhakaran, V., Thijm, T. A., Hirst, G., Stein, B.: Computational argumentation quality assessment in natural language. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, vol. 1, pp. 176–187 (2017)
26. Yang, J., Du, X., Hung, J. L., Tu, C. H.: Analyzing online discussion data for understanding the student's critical thinking. Data Technologies and Applications, vol. 56, no. 2, pp. 303–326 (2022)

ISSN 1870-4069

Assessing the Impact of Data Augmentation on Photovoltaic Module Faults Detection Using Deep Learning Models

Luis E. Montañez[1], Luis M. Valentín-Coronado[1,4], Daniela Moctezuma[2], Diego A. Mercado-Ravell[3,4]

[1] Centro de Investigaciones en Óptica, Aguascalientes, Mexico

[2] Centro de Investigación en Ciencias de Información Geoespacial A. C., Aguascalientes, Mexico

[3] Centro de Investigación en Matemáticas, Zacatecas, Mexico

[4] Consejo Nacional de Humanidades Ciencias y Tecnologías, Ciudad de México, Mexico

{montanezlef, luismvc}@cio.mx, dmoctezuma@centrogeo.edu.mx, diego.mercado@cimat.mx

Abstract. Accurately identifying photovoltaic (PV) module failures is critical to ensuring their reliability and efficiency. Deep Neural Networks (DNNs) have emerged as a highly effective tool for this purpose, particularly when utilizing infrared images. However, it is important to note that optimal DNN performance commonly requires a substantial amount of high-quality and annotated data. Data augmentation techniques have been developed to address this challenge. These techniques involve applying transformations such as rotation and flipping to augment the size of a dataset. However, it is essential to acknowledge that using data augmentation methods carries the inherent risk of introducing biases or inaccuracies into the augmented data. In this study, we conducted a comparative analysis of four data augmentation methods. Our primary objective was to assess the impact of data augmentation on DNN performance through the k-fold cross-validation technique in the PV module failure classification task. Furthermore, we have also tested the generalization capabilities of five different DNN models when utilizing data augmentation methods. This analysis provides valuable insight into the most effective data augmentation methods for enhancing DNN performance and ensuring the accuracy of PV module failure classification. This analysis highlights that the way one uses data augmentation is critical to achieving realistic and reliable results and, hence, reliable models.

Keywords: Infrared images, data augmentation, CNN, classification, photovoltaic modules.

1 Introduction

Photovoltaic (PV) systems are the primary technology utilized to harness solar energy and, in recent years, have gained tremendous popularity as a renewable energy source. The installed PV capacity exhibits an average annual growth rate of 15%. In the past decade, there has been a significant cost reduction of over 40% in installing PV plants. This reduction has resulted in the emergence of more plants and a rise in their size. However, PV plants, while a great source of renewable energy, require regular maintenance and inspections, which can be inconvenient.

The scientific community has explored various options and technologies to decrease the amount of time needed for inspections and enhance their accuracy. One innovative technique that has emerged is using unmanned aerial vehicles (UAVs) equipped with cameras capable of capturing images in both visible and infrared spectrums, also known as thermal images. Aerial inspections performed through this method have been particularly useful for inspecting PV installations. In this sense, thermographic inspection, which involves using infrared imaging to identify defective modules in PV installations, has gained popularity as a reliable and efficient tool [5].

Thermography allows the identification of temperature distributions in PV modules, enabling the detection of non-uniform distributions. Based on this temperature distribution, it is possible to correlate different problems. Nevertheless, a careful image analysis over a large set of infrared images is needed to perform this identification task. Then, to address this challenge, intelligent and/or autonomous systems have been proposed as possible solutions. These include statistical analysis and, more recently, deep learning-based approaches such as Deep Neural Networks in particular Convolutional Neural Networks (CNN) [7, 9, 10, 12].

Despite the large number of CNN models and their different complexities, reviewing the impact of the information used to train and test these networks is necessary. One of the initial strategies for identifying defective modules involved statistical studies of temperatures in infrared images was demonstrated by Kim et al. [6]. In their work, local and global standard deviations were considered to define defective modules. Additionally, Dotenco et al. [2] proposed a statistical analysis of pixel intensities. Carletti et al. [1] introduced another method, employing a water-filling algorithm for hotspot detection. While these methods showcase innovative approaches, traditional techniques face significant challenges in real-world scenarios.

These methods often lack robustness, struggle to generalize to new, unseen data, and encounter difficulties adapting to variations in image conditions, such as distortions, occlusions, or other unpredictable factors. Deep learning, specifically CNNs, has been used as an alternative solution to the PV module image classification problem. The CNNs have demonstrated significant advantages in handling diverse, large-scale, and complex image classification tasks. In some cases, custom convolutional neural models have been proposed [4, 14, 16]. Also, adaptations and transfer learning to re-train well-known models such as VGG-16 [13], MobileNet [3], AlexNet [7], ResNet [9] and DenseNet [12] have been employed. Datasets, some of which are public, are used for training these models. However, one difficulty with current public databases is their limited size and the imbalance in the number of images per class.

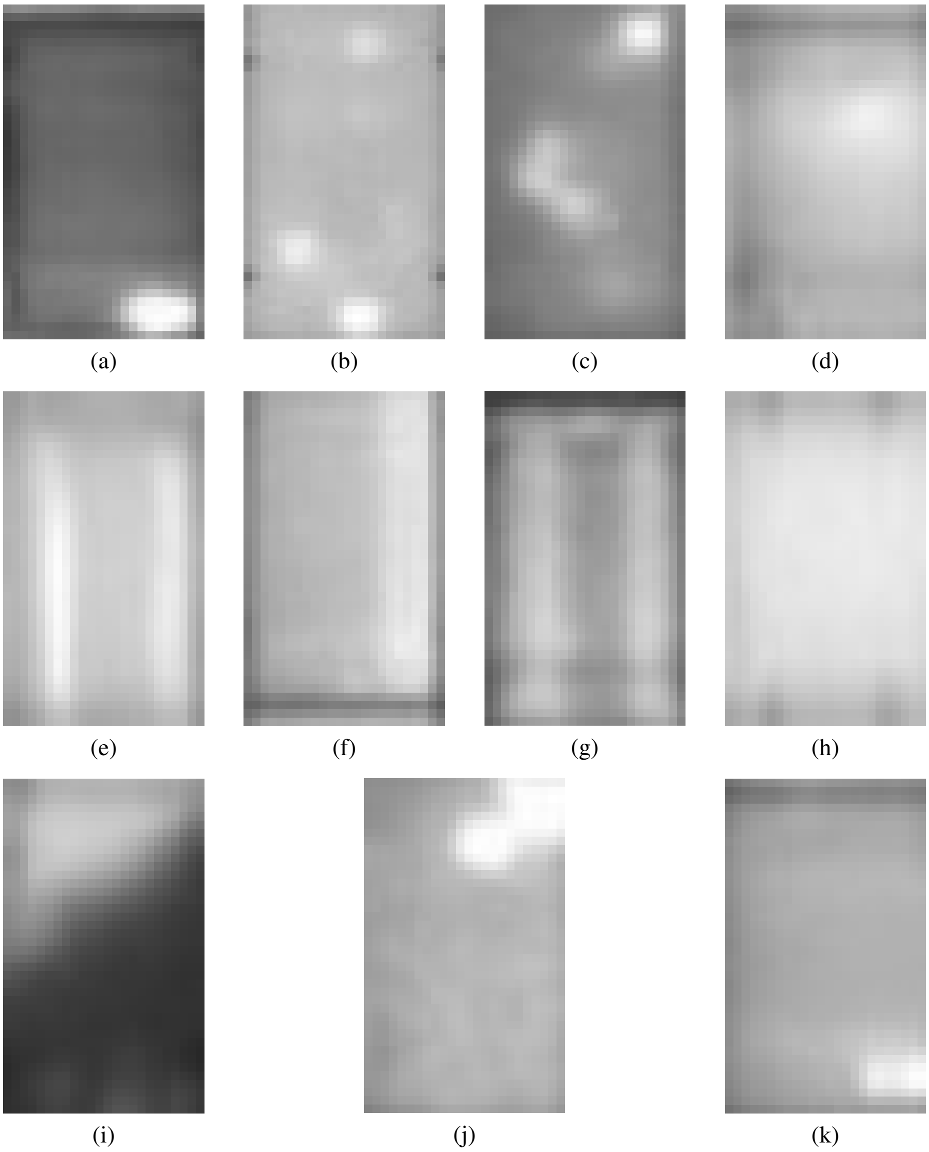


Fig. 1. Example of an image of each class in Millendorf dataset [11]. a) Cell. b) Cell multi. c) Cracked. d) Hotspot. e) Hotspot multi. f) Diode. g) Diode multi. h) Offline. i) Shadowing. j) Soiling. k) Vegetation.

One strategy commonly employed to address these problems is the technique known as data augmentation. Data augmentation consists of applying modifications to existing data, thus creating new instances (synthetic ones) with variations on the original data. In the case of images, geometric transformations such as flipping and rotation operations and brightness modifications have been implemented to generate synthetic images, as reported in the work of Korkmaz [7].

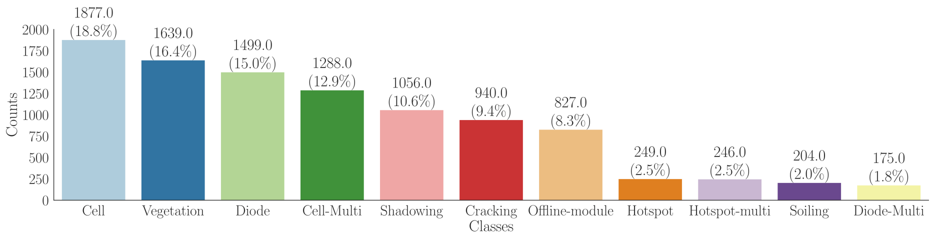


Fig. 2. Distribution of images per class in the dataset [11].

Other works, such as the one presented by Pamungkas et al. [12], use Generative Adversarial Network (GAN) combined with geometric transformations for image generation. However, this combined strategy was outperformed when only geometric data augmentation was used. In this work, we have conducted a comparative analysis of the data augmentation methodologies that some recent works, that address the identification of failures in PV systems, have implemented.

Specifically, we have evaluated the data augmentation methodology of Alves [4], Korkmaz [7], Le [9], and Pamungkas [12], which, in addition, are using the database reported by Millendorf et al. [11]. The analysis performed here primarily aims to assess the impact of data augmentation on these works, since they have reported very different results depending on data augmentation strategies besides the model employed. To ensure a fair comparison, we implemented some of the most recognized and widely accessible deep neural network architectures, modifying only how data augmentation was implemented.

The contribution of this analysis is the evaluation of the effectiveness of different data augmentation techniques to improve the performance of deep learning models in the detection of faults in photovoltaic systems. The remainder of this manuscript is organized as follows. In Section 2 the proposed methodology for comparing training and data augmentation methodologies found in the literature is presented. Section 3 describes the obtained results and their analysis. Finally, in Section 4, the conclusions and future work are presented.

2 Methodology

To assess the impact of data augmentation methodologies, we propose using transfer learning for five well-known deep learning models in the literature: VGG16, ResNet50, MobileNetV2, DenseNet121, and EfficientNetB0. Using widely recognized models, which have proven effective in various classification tasks and can be easily reproduced, the aim is to establish a solid basis for evaluating the proposed methodology. These classification models come with pre-trained weights on the ImageNet dataset [8].

Additionally, the classification models undergo a modification in the last output layer to classify the number of classes of failures present in the thermal image dataset. The employed dataset in this analysis comprises $10,000$ thermal aerial images of PV modules, each measuring 24×40 pixels, with eleven different failure classes. These grayscale images were acquired during various infrared aerial inspections and manually classified by Millendor et al. [11].

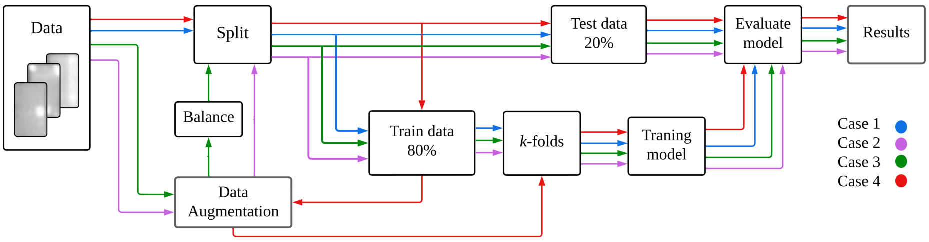


Fig. 3. Proposed methodology to address four cases of data augmentation. Here, k-fold cross-validation is implemented during the training stage of the CNN models. The best-performing model from the k-fold cross-validation is used to evaluate the models.

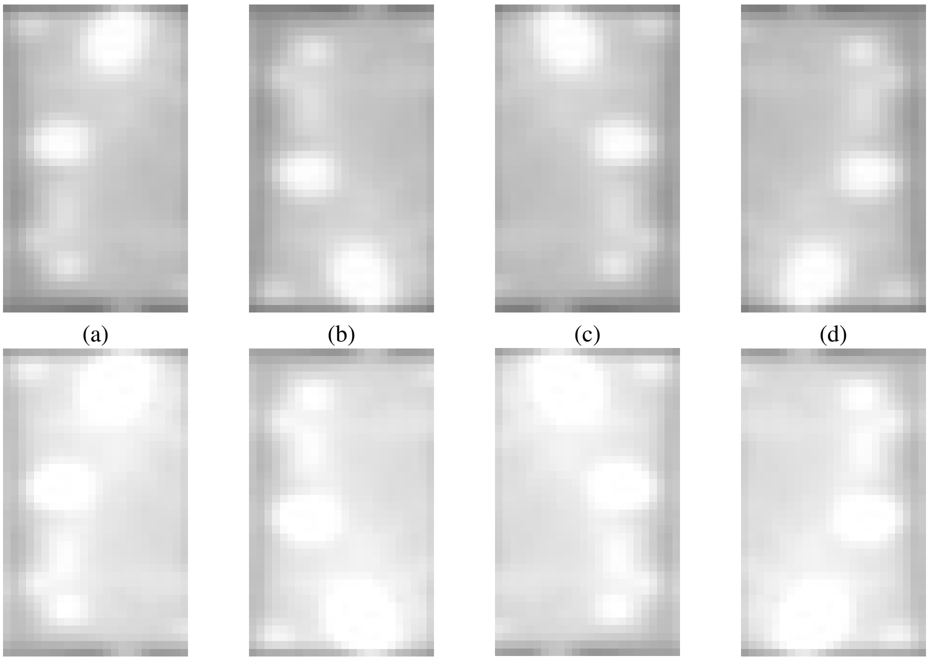
Depending on the type of failure, each PV module image shows a different thermal distribution, where areas with higher temperatures are highlighted in bright colors, while areas with lower temperatures are highlighted in dark colors. An image for each class is shown in Fig. 1. Due to varying fault frequencies, the dataset exhibits an imbalance in the number of images per class. The classes are Cell, Cell multi, Cracked, Hotspot, Hotspot multi, Diode, Diode multi, Offline, Shadowing, Soiling and Vegetation. Cell failure has the highest incidence at 18.8%, while multiple diode bypass, soiling, hotspot, and multiple hotspots represent the lowest incidence classes at 1.8%, 2%, 2.5%, and 2.5%, respectively.

The distribution of images in each failure class is depicted in Fig. 2. Data augmentation is employed to address imbalance issues. Based on current literature, we have proposed a methodology that incorporates four distinct data treatments: i) No data augmentation, ii) fully augmented data set, iii) data augmentation and post-augmentation class balance, and iv) data augmentation only in the training set. Furthermore, a k-fold cross-validation with $k = 10$ is implemented during the training model stage in all cases.

The k-fold cross-validation involves partitioning the training subset into k distinct folds, where one fold serves as the validation dataset while the remaining folds collectively form the training set of the model. This process is iterated k times, with each fold acting as the validation subset. Using k-fold cross-validation provides a robust metric for assessing the model's generalization capabilities.

The whole methodology is illustrated in Fig. 3, where each color line represents each previously described case of study. In the first case, the dataset is used without augmentation, ignoring the imbalance in the number of images per class, as used in the works by Alves-Fonseca [4], Li [10], Le [9] and Pamungkas [12]. Then, the dataset is split into two subsets: 80% for training and 20% for testing.

In the second case, data augmentation is applied, including flipping, rotation, and brightness operations to increase the number of samples as described in the work of Pamunkgas [12]. Brightness operations involve decreasing and increasing each pixel intensity value by ± 30. Subsequently, flipping and rotation operations are applied to original and brightness-modified images. This augmentation generates 11 additional images per original image. Fig. 4 shows representative examples of the generated synthetic images. The resulting set, which comprises 120,000 images with the original class imbalance, is split into two subsets: 80% for training and 20% for testing.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Example of data augmentation including geometric and brightness operations. a) Original image. b) Vertical flip. c) Horizontal flip. d) 180 degrees rotation. e) Brightness increase. f-h) Vertical, horizontal, and 180 degrees rotation with brightness increase. i) Decrease of brightness. j-l) Vertical, horizontal and 180 degrees rotation with a decrease in brightness.

Similar to the second case, data augmentation is carried out using the same geometric operations and brightness modifications for the third case. However, unlike case two, a data balance is performed, ensuring each class has an equal number of elements. Uniformly random image selection of the augmented dataset is performed to balance the classes. Each class contains the same number of images as the smallest class in the augmented dataset. Then, 1,100 images from each class were uniformly randomly chosen from a final set of 120,000 images.

These images are divided into training and test subsets following an 80-20 proportion. In the fourth scenario, data augmentation is exclusively implemented on the training subset. Then, a training set of 96,000 images is obtained. It is worth mentioning that for all cases, the best-performing model from the k-fold cross-validation is tested on the dedicated test subset. The overall performance of the models has been evaluated using widely known metrics such as Accuracy, Precision, Recall, and F_1-score [15].

3 Results

In this section, the results obtained from the proposed methodology are presented. A desktop computer with Core i9 processor, NVIDIA GPU GeForce GTX 3090 with 24 GB of VRAM, and 64 GB of RAM was used to extract features and train the models.

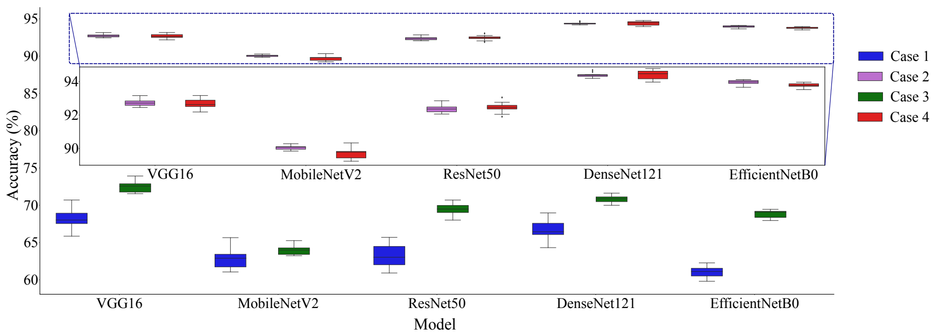


Fig. 5. Comparison of the training performance, after the 10-fold cross-validation implementation, of each of the four cases using the five deep neural networks models.

The implementation was carried out in Python 3.8 and PyTorch framework 1.13.1. In all four cases, the AlexNet model was trained using the Adam optimizer on 25 epochs with a learning rate of 0.0001, a batch size of 32, and categorical cross-entropy as a loss function. For each of the models used in the fine-tuning process, a linear layer of the same size as the last layer of the original network was added, specifically adjusted to match the number of classes required in the classification task.

Fig. 5 provides a graphical representation of the accuracy distribution obtained by each model (VGG16, ResNet50, MobileNetV2, DenseNet121, and EfficientNetB0) in the four different data augmentation cases. The data presented in Fig. 5 indicates an improvement in the performance of all deep neural network models when data augmentation is utilized (cases 2, 3, and 4) compared to the scenario where no data augmentation is applied (Case 1). However, it is worth highlighting that the accuracy reaches only about 75% when data augmentation and post-augmentation class balance are employed (Case 3). This may be attributed to the fact that data sampling is performed after augmentation, which may result in an inadequate representation of each class during the model training phase.

It is crucial to mention that this issue is observable regardless of the model type. Furthermore, it can be observed that the measure of central tendency is higher for Cases 2 and 4 while also presenting a comparatively lower measure of dispersion. The average accuracy values and their corresponding standard deviations obtained through 10-fold cross-validation for each model in four training cases are shown in Table 1. It stands out that DenseNet121 achieved the highest average accuracy, reaching a remarkable 94.4% in Case 4. Also, it is observed that EfficientNetB0 exhibited the lowest standard deviation in Case 4.

In Case 1, the performance is less than 70% in accuracy, so it can be suggested that the imbalance and the small number of images affect the generalization ability. Since the standard deviation is greater than 1 in most cases, it can be concluded that there is significant variability depending on the fold employed. By employing data augmentation in Case 2, an increase of almost 28% in average accuracy is observed, while maintaining a standard deviation of less than 0.24. It is crucial to note that including images to balance the classes (Case 3) leads to a further increase in the average accuracy, as evidenced by comparing Case 1 with Case 3.

Table 1. Mean accuracy (μ_A) and standard deviation (σ_A) in the training of k-folds, presented as percentages.

	μ_A				σ_A			
Model	**Case 1**	**Case 2**	**Case 3**	**Case 4**	**Case 1**	**Case 2**	**Case 3**	**Case 4**
DenseNet121	66.62	**94.39**	70.83	**94.4**	1.36	0.15	0.56	0.28
EfficientNetB0	61.05	93.95	68.82	93.76	0.76	0.15	0.53	0.14
MobileNetV2	62.78	90.06	64.01	89.71	1.38	0.13	0.7	0.32
ResNet50	63.28	92.37	69.44	92.45	1.72	0.23	0.81	0.33
VGG16	**68.3**	92.75	**72.49**	92.67	1.42	0.23	0.78	0.32

Table 2. Classification report showing the percentage performance of the best models on the test subset across four different cases.

Model	**Case 1**					**Case 2**				
	Acc	**P**	**R**	**F1**	**#**	**Acc**	**P**	**R**	**F1**	**#**
VGG16	69.40	**69.03**	63.89	**65.78**	2,000	95.41	94.74	95.02	94.86	13,000
ResNet50	60.25	63.13	51.31	52.69	2,000	97.38	97.08	97.47	97.25	13,000
DenseNet121	**70.35**	67.77	**64.25**	65.40	2,000	97.72	97.73	97.81	97.76	13,000
MobileNetV2	67.05	63.18	59.85	60.84	2,000	94.50	94.60	94.68	94.61	13,000
EficientNetB0	67.15	65.53	63.09	63.76	2,000	**98.52**	**98.78**	**98.54**	**98.52**	13,000
Model	**Case 3**					**Case 4**				
	Acc	**P**	**R**	**F1**	**#**	**Acc**	**P**	**R**	**F1**	**#**
VGG16	**76.58**	**76.09**	**76.58**	**76.58**	4,620	73.90	69.51	67.86	68.31	2,000
ResNet50	74.65	75.22	74.65	74.60	4,620	73.30	72.33	69.28	69.89	2,000
DenseNet121	75.84	75.31	75.84	75.35	4,620	73.25	**73.75**	70.06	**70.82**	2,000
MobileNetV2	66.34	66.25	66.34	66.11	4,620	73.55	71.96	**70.46**	70.79	2,000
EficientNetB0	74.31	74.61	74.31	73.53	4,620	**74.55**	71.75	69.94	70.17	2,000

The differences between Case 2 and Case 4 do not seem to be significant, at least in terms of performance during training.Through cross-validation, the model with the highest accuracy was identified. This model was tested using the test subset, and the results of the best models are detailed in Table 2. In Case 1, the top models were VGG-16 and DenseNet, achieving accuracy close to 70%, with values above 64% in the precision, recall, and F_1 metrics. In Case 2, the most outstanding model was EfficientNetB0, achieving values above 98% in all metrics. For the third case, VGG-16 proved to be the best model with a performance of 76% in all metrics.

DenseNet121 emerged as the best model for the fourth case, presenting two of the highest values in the accuracy and F_1-score metrics, with 73% and 70.82%, respectively. The notably high values are highlighted in bold in Table 2. It can be seen that, for Case 2, all metrics exceed 94%, while for the other cases, they do not reach the average value of 77%. This analysis highlights the remarkable differences in the performance of the models in different training configurations. In cases 1, 2, and 3, no significant differences are observed between the accuracy values reported during training and those obtained in the test phase.

Table 3. Comparison of works employing the dataset reported by Millendorf et al. [11].

Reference	Case	Model	DATe	DATr	Balance	Accuracy
Alves Fonseca 2021 [4]	1	CNN	✗	✗	✗	66.43
Le 2021 [9]	1	Ensamble	✗	✗	✗	85.9
Li 2023 [10]	1	Transformer	✗	✗	✗	**88.5**
Pamungkas 2023 [12]	1	UDenseNet	✗	✗	✗	65.9
This analysis	1	DenseNet121	✗	✗	✗	70.35
		VGG16	✗	✗	✗	69.40
Pamungkas 2023 [12]	2	UDenseNet	✓	✓	✗	96.65
This analysis	2	EfficientNetB0	✓	✓	✗	**98.78**
Korkmaz 2022 [7]	3	CNN Multiscale	✓	✓	✓	**93.51**
This analysis	3	VGG-16	✓	✓	✓	76.58
This analysis	4	EfficientNetB0	✗	✓	✗	**74.55**

However, in Case 4, significant discrepancies of at least 20% are evident. This disparity can be attributed to the fact that the data augmentation in Case 4 is performed exclusively on the training set, after its separation from the test set. As a consequence, the test set contains images that the model has never previously encountered, which may influence the evaluation metrics. This phenomenon suggests that by employing data augmentation on the dataset intended for testing, the illusion could be generated that the model has exceptional generalization ability, when in fact it may be suffering from overfitting to specific patterns present in the training set.

Contrarily, it is found that applying data augmentation exclusively on the training set (Case 4) results in a 4% increase in generalization ability compared to Case 1. This supports the notion that data augmentation on the training set can have a positive impact on the model's ability to generalize to previously unseen data, even when faced with a test set with unpublished images. In addition, a comparison was made between the results obtained in this study and those reported in the literature for models with the best accuracy values in each specific case shown in Table 3. Accuracy is the only metric employed, since the state of the art works only report this metric. The findings are as follows:

– Case 1. The Vision Transformer-based model proposed by Li et al. [10] stands out, outperforming conventional Convolutional Neural Networks (CNN), despite the class imbalance. This result suggests that, under imbalance conditions, transformer-based architectures can offer better generalization capability compared to conventional CNNs.

– Case 2. The EfficientNetB0 model outperforms the model of Pamungkas et al. [12] by almost 3%, even though the model used is less complex. It is important to note that higher accuracy does not always reflect the true generalization ability of a model, as model complexity also plays a crucial role.

Table 4. Comparison of training time of each model.

	Time by model (min)					
Case	**VGG16**	**MobileNet**	**ResNet**	**DenseNet**	**EfficientNet**	μ
I	15	5	9	10	8	9.4
II	235	94	146	158	128	152.2
III	37	14	23	25	20	23.8
IV	189	72	116	149	105	126.2

– Case 3. The model reported by Korkmaz et al. [7] shows exceptional performance, reaching a value of 93.51%, significantly outperforming the best model proposed in this study. However, a discrepancy in the reported accuracy is observed, as the work of Li et al. [10] when replicating the model, obtained an accuracy of 85.1% employing the Case 1 method. These discrepancies highlight the importance of reproducibility and consistency in presenting results in the scientific literature.

– Case 4. By implementing data augmentation exclusively on the training set, a 4% increase in generalization ability is observed compared to Case 1, although it is still not optimal classification. These results emphasize the importance of exploring additional approaches or making adjustments to improve the generalization ability substantially.

Additionally, Table 4 presents the time required for 10-fold cross-validation in the training stage for each model and the four data augmentation cases. As it can be observed, in Case 1, no model exceeded 15 minutes of training. In contrast, Case 2 showed a mean training time of 152.2 minutes, where the VGG16 model achieved a maximum of 235 minutes, whereas the MobileNet achieved the minimum time (94 minutes). Note that Case 2 denotes the scenario with the highest training time.

It is important to note that the relationship between the number of images used and the resulting training time is not linear. When comparing Case 1 and Case 2, in which the number of images used differs by a factor of 12 due to increased data, the training time increases by at least 15 times. In contrast, when Case 1 and Case 3 are compared, where the number of images used differs by a factor of 2.3, the training time increases by at least 2.4 times. Then, it demonstrates that a linear increase in the number of images used does not result in a linear increase in training time. Among all the implemented models, VGG-16 consistently exhibited the longest training times, whereas MobileNetV2 consistently demonstrated the shortest training times.

This discrepancy can be attributed to the substantial contrast in the number of trainable parameters. VGG-16 possesses 138.4 million parameters, whereas MobileNetV2 possesses only 3.5 million parameters, amounting to approximately 40 times fewer parameters. In summary, it has been observed that data augmentation can be a highly effective technique to enhance the performance of deep learning models. For instance, in the work of Pamungkas [12], it was found that augmenting the training and testing sets significantly improved model performance by approximately 30%, compared to no data augmentation.

However, it is important to note that this performance may be biased due to the incorporation of prior knowledge (augmented data) in the testing process, which could lead to overfitting and reduce the model's generalization capability. Nevertheless, it is possible to improve model performance even with data augmentation limited to the training set, as shown by the performance improvement of the EfficientNetB0 model in Case 4 compared to Case 1.

4 Conclusions

In this study, the analysis of the impact on the performance of several data augmentation methodologies reported in the literature was done. Data augmentation was found to be a valuable tool for improving the generalization capability of the model, as well as for achieving a balance in the class distribution. Nonetheless, what was emphasized was that the application of these techniques on both training and test datasets may generate the illusion of an improvement in classification ability.

Crucially, this apparent improvement does not necessarily guarantee the generalization capacity of the model facing new data. In addition, it should be noted that the increase in the number of images used for training prolongs the time required, although this increase does not follow a linear relationship. As a perspective for future work, a more exhaustive exploration of models based on Transformers is contemplated, exploring their analysis and evaluation in more detail. Additionally, exploring more advanced data augmentation techniques, such as synthetic data generation using Generative Adversarial Networks (GANs), is also considered.

Acknowledgments. Thanks to the scholarship 05579 of Becas Nacionales provided by CONAHCYT.

References

1. Carletti, V., Greco, A., Saggese, A., Vento, M.: An intelligent flying system for automatic detection of faults in photovoltaic plants. Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 5, pp. 2027–2040 (2019) doi: 10.1007/s12652-019-01212-6
2. Dotenco, S., Dalsass, M., Winkler, L., Würzner, T., Brabec, C., Maier, A., Gallwitz, F.: Automatic detection and analysis of photovoltaic modules in aerial infrared imagery. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, pp. 1–9 (2016) doi: 10.1109/WACV.2016.7477658
3. Dunderdale, C., Brettenny, W., Clohessy, C., van-Dyk, E. E.: Photovoltaic defect classification through thermal infrared imaging using a machine learning approach. Progress in Photovoltaics: Research and Applications, vol. 28, no. 3, pp. 177–188 (2019) doi: 10.1002/pip.3191
4. Fonseca-Alves, R. H., Deus-Júnior, G., Marra, E. G., Lemos, R. P.: Automatic fault classification in photovoltaic modules using convolutional neural networks. Renewable Energy, vol. 179, pp. 502–516 (2021) doi: 10.1016/j.renene.2021.07.070
5. Grimaccia, F., Leva, S., Dolara, A., Aghaei, M.: Survey on pv modules' common faults after an O&M flight extensive campaign over different plants in italy. IEEE Journal of Photovoltaics, vol. 7, no. 3, pp. 810–816 (2017) doi: 10.1109/JPHOTOV.2017.2674977

6. Kim, D., Youn, J., Kim, C.: Automatic photovoltaic panel area extraction from UAV thermal infrared images. Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, vol. 34, no. 6, pp. 559–568 (2016) doi: 10.7848/ksgpc.2016.34.6.559
7. Korkmaz, D., Acikgoz, H.: An efficient fault classification method in solar photovoltaic modules using transfer learning and multi-scale convolutional neural network. Engineering Applications of Artificial Intelligence, vol. 113, pp. 104959 (2022) doi: 10.1016/j.engappai.2022.104959
8. Krizhevsky, A., Sutskever, I., Hinton, G. E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM, vol. 60, no. 6, pp. 84–90 (2017) doi: 10.1145/3065386
9. Le, M., Luong, V. S., Nguyen, D. K., Dao, V. D., Vu, N. H., Vu, H. H. T.: Remote anomaly detection and classification of solar photovoltaic modules based on deep neural network. Sustainable Energy Technologies and Assessments, vol. 48, pp. 101545 (2021) doi: 10.1016/j.seta.2021.101545
10. Li, S., Chen, H., Zhang, A., Gong, C., Menhas, M. I., Liang, W., Wang, Z., Yang, N.: Photovoltaic panel fault detection and diagnosis based on a targeted transformer-style model. IEEE Transactions on Industry Applications, vol. 60, no. 1, pp. 1814–1826 (2024) doi: 10.1109/TIA.2023.3322688
11. Millendorf, M., Obropta, E., Vadhavkar, N.: Infrared solar module dataset for anomaly detection. In: International Conference on Learning Representations, pp. 1–5 (2020)
12. Pamungkas, R. F., Utama, I. B. K. Y., Jang, Y. M.: A novel approach for efficient solar panel fault classification using coupled udensenet. Sensors, vol. 23, no. 10, pp. 4918 (2023) doi: 10.3390/s23104918
13. Pierdicca, R., Malinverni, E. S., Piccinini, F., Paolanti, M., Felicetti, A., Zingaretti, P.: Deep convolutional neural network for automatic detection of damaged photovoltaic cells. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII–2, pp. 893–900 (2018) doi: 10.5194/isprs-archives-xlii-2-893-2018
14. Po-Ching-Hwang, H., Cheng-Yuan-Ku, C., Chi-Chang-Cha, J.: Detection of malfunctioning photovoltaic modules based on machine learning algorithms. IEEE Access, vol. 9, pp. 37210–37219 (2021) doi: 10.1109/ACCESS.2021.3063461
15. Vujovic, Z. D.: Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, vol. 12, no. 6, pp. 599–606 (2021) doi: 10.14569/ijacsa.2021.0120670
16. Zefri, Y., Sebari, I., Hajji, H., Aniba, G.: Developing a deep learning-based layer-3 solution for thermal infrared large-scale photovoltaic module inspection from orthorectified big UAV imagery data. International Journal of Applied Earth Observation and Geoinformation, vol. 106, pp. 102652 (2022) doi: 10.1016/j.jag.2021.102652

ISSN 1870-4069

Automatic COPD Detection through Vocal Emissions Using Intelligent Audio Analysis

Carlos Leyva-Bravo[1], Ivan Gallardo-Bernal[2], Humberto Pérez-Espinosa[1]

[1] Universidad Autónoma de Guerrero, Maestría en Tecnologías de la Información, Mexico

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

carosleyvabravo@gmail.com, humbertop@inaoep.mx, gallardo@uagro.mx

Abstract. This study presents an application of artificial intelligence for the early detection of chronic obstructive pulmonary disease (COPD) through analyzing vocal emissions. Using audio processing techniques and machine learning, we analyzed 30 vocal recordings from individuals in vulnerable communities in Guerrero, Mexico. These recordings included clinically confirmed COPD patients, healthy controls, and individuals exposed to smoke from burning organic materials. Our approach employs a Transformer-based classifier to analyze vocal patterns and identify distinctive COPD characteristics, providing a non-invasive and accessible screening tool. Implementing AI in this context addresses significant healthcare access disparities by offering a scalable and cost-effective diagnostic tool for communities with limited access to advanced technologies. Preliminary results indicate that our transformer-based model effectively distinguishes between COPD patients and other groups, demonstrating its potential to enhance early detection and improve patient outcomes. This study underscores the transformative impact of artificial intelligence in promoting health equity and advancing public health in marginalized populations.

Keywords: COPD, respiratory diseases screening, intelligent audio analysis.

1 Introduction

Chronic Obstructive Pulmonary Disease (COPD) is a severe medical condition that critically impairs the respiratory system, significantly limiting the ability to breathe. According to the World Health Organization (WHO), COPD was the third leading cause of death worldwide in 2023, responsible for more than 3 million deaths [5]. Characterized by persistent obstruction of airflow in the lungs, COPD leads to breathing difficulties and substantially decreases quality of life.

The development of COPD is strongly associated with prolonged exposure to pulmonary irritants, smoking being the main risk factor. However, exposure to environmental pollutants, dust and chemicals also contributes to its onset [7]. COPD typically encompasses two main conditions: chronic bronchitis and emphysema, which often co-occur, exacerbating the disease's severity and the extent of lung damage in affected individuals. The situation is exacerbated in areas with high social marginalization and rural regions in Mexico, such as the southern and southeastern states such as Oaxaca, Guerrero, Chiapas, and Michoacán, where biomass (organic material derived from plants and animals) is a common energy source for cooking and heating homes [10]. The use of biomass contributes to poor air quality and increased respiratory health issues.

According to the National Institute of Statistics and Geography (INEGI) [11], COPD ranks tenth among the leading causes of death in these regions. In the state of Guerrero, chronic obstructive pulmonary diseases were the 11th leading cause of mortality between 2020 and 2022, as reported by the Epidemiological and Statistical Death System (SEED). The population of Guerrero is diverse, comprising various indigenous groups. However, they face significant vulnerabilities, including poverty, marginalization, discrimination, and limited access to basic health services due to inadequate infrastructure, a shortage of medical personnel, and difficulties in accessing medical care. These factors heighten the risk of poor health outcomes, underscoring the need for targeted interventions to improve healthcare access and equity.

2 Related Work

Studies on COPD have revealed notable differences between patients exposed to biomass and those affected by tobacco. Symptoms such as dyspnea, chronic bronchitis, rales, and wheezing were more common in the biomass group [3]. Spirometric measurements showed higher levels of severity of Forced Expiratory Volume in 1 Second (FEV1) and Forced Vital Capacity (FVC) in patients exposed to biomass compared to tobacco exposure. Radiographically, emphysema was more prevalent in the tobacco group, while bronchiectasis and atelectasis were more common in patients exposed to biomass. Another study [2] analyzed the effects of indoor biomass smoke pollution, concluding that biofuel smoke adversely affects the respiratory system. It affects the lung parenchyma and contributes to the burden of respiratory diseases, including COPD and tuberculosis, in exposed populations.

Artificial intelligence is transforming various medical specialties, with notable advances in neurology, oncology, radiology, and clinical pathology [8]. For instance, AI-based algorithms in oncology accurately diagnose cancer in computational histopathology. In radiology, deep learning software shows significant strides in image-based diagnosis compared to radiologists. Integrating AI in the diagnosis and management of COPD marks a substantial advance in contemporary medicine [9]. AI can reduce the global burden of COPD, reduce healthcare costs, and improve early diagnosis [19]. Improves clinical data interpretation, facilitating early intervention and more effective COPD management. In a recent study [21], the authors reviewed the role of AI in diagnosing and treating respiratory diseases such as COPD and asthma.

The findings highlight the utility of AI in clinical, functional, and imaging analysis and in developing clinical prediction models and remote patient monitoring. AI for respiratory disease analysis, such as the eRx tool [6] with its CNNs algorithms, utilizes innovative techniques for remote X-ray analysis in suspected tuberculosis cases. [15] investigated the early detection of respiratory diseases using advanced signal processing and machine learning [1]. The study achieved notable accuracy in detecting respiratory events such as crackles and wheezing through autoregression combined with SVM and CNN. CNNs for image analysis and classification algorithms for structured data were used in [20]. The final model, which integrated information from X-rays and clinical data, achieved high accuracy in the detection of COPD, which is a valuable tool to improve medical diagnosis and influence future research in respiratory health.

AI can identify early signs of diseases such as COPD, asthma, pneumonia, and COVID-19 [18]. by analyzing vocal emissions, such as cough contents and lung sounds [13]. This early detection capability is crucial for timely diagnosis and effective treatment, improving patient outcomes and quality of life. For example, a machine listening system was developed using deep learning to detect coughs and diagnose early respiratory diseases from noisy audio signals [17]. The system employed logarithmic spectrograms and a convolutional neural network, achieving high sensitivity.

We found limitations in the reviewed related works that our proposal aims to address. One of them is the lack of diversity and breadth in datasets, affecting the generalization and accuracy of diagnoses. In addition, challenges in external validation raise doubts about the reliability of previous findings. Our proposal addresses these issues by integrating diverse data sets and employing advanced AI and machine learning techniques. We prioritize rigorous validation through clinical studies and real-world testing, ensuring our solution's reliability across various medical contexts. By overcoming these limitations, our proposal significantly improves the early detection and management of COPD, advancing medical research.

3 Methodology

A dataset was constructed to train a COPD detection model based on audio recordings obtained from lung sounds, including cough recordings from individuals from the most relevant indigenous communities inhabiting the state of Guerrero with COPD and healthy subjects as the sample population. The study included three groups of patients, each consisting of 10 individuals from vulnerable communities in the state of Guerrero. The first group comprised healthy individuals without respiratory diseases or chronic conditions. The second group was formed by 10 individuals with suspected COPD. The third group included 10 individuals clinically diagnosed with COPD.

Sample collection was carried out in clinics of the Mexican Social Security Institute located in the state of Guerrero, where patients attended check-up consultations and complications in respiratory diseases. Data collection was supervised at all times by the specialist physicians and nurses on duty. In addition, a meticulous and ethical process of collecting personal, hereditary, and general health data was carried out, with special attention to their self-determination as Indigenous people, delivering to each patient an informed consent detailing the use and responsible management of this information for academic and research purposes. In addition to the above, each patient was provided with a questionnaire based on the GOLD (Global Initiative for Chronic Obstructive Lung Disease) dyspnea scale to assess respiratory difficulty. In addition, relevant demographic data of the studied population were collected.

4 Capture of the Sample

The study included participants from the Amuzgos, Mixtecos, Tlapanecos, and Nahuas communities. The first group comprised 7 Amuzgos patients, the second group included 8 Mixtecos patients, the third group had 8 Tlapanecos patients, and the fourth group consisted of 9 Nahuas patients. For the study, a total of 420 cough data from patients in indigenous communities were used to classify between positive and negative cases of COPD. The data were divided into training sets (70%) and test sets (30%), stored in CSV files. An effort was made to maintain a representative proportion of positive and negative patients in both data sets to ensure that the classification model correctly learned the differences between the classes.

In the training set, having a larger amount of data allows the model to have more examples to learn from, thus improving the model's ability to generalize to unseen data. The testing set should be large and representative enough to evaluate the model's performance accurately. Using 126 samples for testing ensures that the model is not overfitting to the training data and that its performance can be evaluated on unseen data. Class balance in the training set is essential to avoid bias towards one class. With 7 positive patients and 14 negative ones, a certain balance is ensured, which helps the model effectively learn to distinguish between both classes.

In the testing set, maintaining a proportion of positive patients helps to evaluate whether the model has a high false negative rate or if it can accurately detect positive cases. Undoubtedly, this model will need to be tested with larger data sets; however, collecting data from patients with COPD is not an easy task, as it is not common for patients to attend regular medical check-ups but rather situations ranging from clinical complications to emergencies. Each CSV file contains relevant information about the audio files, such as the file path and name and the associated class (Positive or Negative). The data were processed to load and resample the audio files to 16 kHz and then used to train and evaluate a classification model. The main formula of the attention mechanism is Scaled Dot-Product Attention. It is defined as:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{1}$$

where: QKT: First, the product of matrices between Q (Query) and the transpose of K (Key) is calculated, resulting in a matrix of similarity scores. Each score indicates how relevant a key is to a query value in the similarity matrix is divided by the square root of the dimension of the keys (Keys). This operation is performed to prevent scores from becoming too large, which could lead to very small gradients during backpropagation (vanishing gradients) [14]. Softmax: The softmax function is applied to the resulting matrix to normalize the scores from 0 to 1. This converts the scores into probabilities, indicating the relative importance of each key (Key) concerning the query (Query).

Multiplication by V: The probability matrix is multiplied by V (value) to obtain the weighted output. This operation combines the values according to their relative importance determined by the attention.

5 Wav2Vec 2.0 Model

Wav2Vec 2.0 is a speech recognition model developed by Facebook AI (now Meta AI) that has proven effective in automatic speech recognition (ASR) tasks. Wav2Vec 2.0 focuses on self-supervised pretraining, [4] which means it can learn robust speech representations using unlabeled data. It can then be fine-tuned with a much smaller amount of labeled data for specific speech recognition tasks, and this model can be adapted using audio signals for the detection of COPD or other respiratory diseases.

The structure of Wav2Vec 2.0 consists of two main parts: The Feature Encoder and the contextual network [12] The Feature Encoder transforms the raw audio signal into more manageable representations using a convolutional network that extracts local features from the audio wave. The signal is divided into small windows to be processed by different convolutional layers, and the output is a sequence of vectors representing the audio signal in a compact and latent form.

The contextual network of Wav2Vec 2.0 uses a Transformer Encoder to analyze speech and capture relationships between different elements in the audio sequence. This component is crucial to understanding the global context of speech [16]. In the Transformer Encoder of Wav2Vec 2.0, self-attention layers allow each latent vector in the sequence to attend to all other vectors. This allows the model to capture long-term dependencies in the audio sequence.

Additionally, each self-attention layer is followed by a feed-forward neural network applied to each latent vector independently. These feedforward networks help transform the representations nonlinearly, enriching the model's ability to capture complex patterns in the data. To stabilize and improve training, each sublayer in the Transformer Encoder of Wav2Vec 2.0 has residual connections and layer normalization. Residual connections help avoid gradient degradation issues and enable more effective training.

6 Data Collection Protocol

For this study, patients were asked to cough for 10 seconds, take a break, and cough again for 10 seconds until they completed a minute of recording the cough with a high-quality microphone. A capture application was developed, where the patients' coughs were recorded to later save them in WAV format for later processing with artificial intelligence methods and techniques.

7 Data Preprocessing

Subsequently, the sound data were preprocessed. The recordings were standardized to 1.4 seconds by repeating or cropping the sound signals. They were labeled as "Positive" (sick) or "Negative" (healthy) and separated into training and testing sets. The recordings were loaded and resampled to a frequency of 16 kHz.

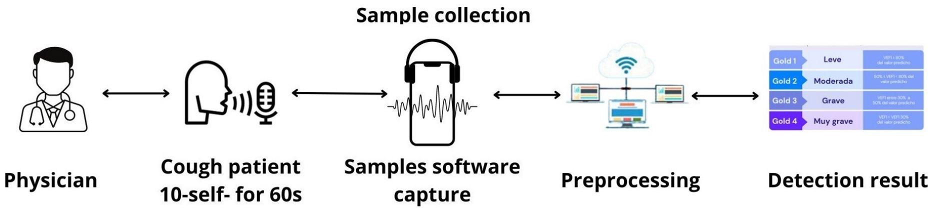


Fig. 1. Data collection protocol. Own creation.

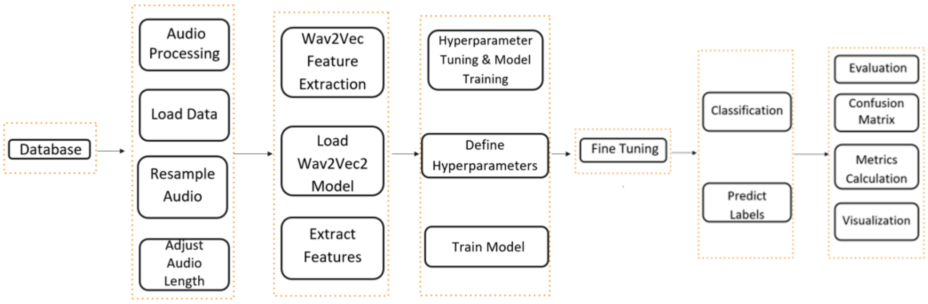


Fig. 2. Wav2Vec 2.0 process stages. Own creation.

8 Feature Extraction

Next, the pre-trained Wav2Vec 2.0 model, designed for voice recognition, was utilized to extract features from the audio recordings. This model transforms the audio signals into latent representations that capture important information for classification. To obtain the final classification, a global pooling layer and additional dense layers were applied to these representations, with regularization through dropout.

9 Optimization and Training

The training was optimized using the Hyperopt library to find the optimal hyperparameters. Finally, the performance of the model was evaluated in the test set, obtaining precision, recall, specificity, accuracy, and F1 score metrics. The results were displayed in precision and loss graphs, and a confusion matrix was generated to analyze the model's performance in COPD detection.

10 Results

Comparing the model's results with the pre-trained database and with the new specific data from the vulnerable population allowed the evaluation of the model's ability to generalize to different datasets. This is crucial to ensure that the model does not overfit a specific dataset and can be applied to diverse populations. The study demonstrates the potential of pre-trained AI models for COPD classification, showing promising results even when applied to data from vulnerable patients in the state of Guerrero.

Table 1. Pre-trained model results.

Results	Precision	Recall	F1-Score	Support
Healthy	0.72	0.77	0.74	30
Sicks	0.89	0.86	0.88	66
Macro avg	0.80	0.82	0.81	96
Weighted avg	0.84	0.83	0.83	96

Table 2. Study results in vulnerable populations.

Results	Precision	Recall	F1-Score	Support
Healthy	0.71	0.72	0.77	10
Sicks	0.88	0.87	0.88	10
Macro avg	0.80	0.81	0.81	20
Weighted avg	0.81	0.82	0.80	20

Despite some variations in performance, the model's ability to adapt to new data underscores its flexibility and utility. Although the model's accuracy varied between the original and new data, these results indicate the model's robustness. The observed decrease suggests specific areas for improvement, providing a solid foundation for future optimizations and adjustments. Differences in the sensitivity and specificity of the model when analyzing patient data highlight the importance of incorporating specific genetic and environmental factors.

These variations provide valuable insights that can be used to customize and improve the model. Training Set Results. The first section of the table shows the results obtained when the model was applied to the training data. These results are broken down by class, that is, "healthy" individuals and "sick" individuals. For each class, the following metrics are reported:

– Precision: The proportion of true positives among the total predicted positives. For the healthy individuals class, the precision was 0.72, while for the sick individuals, it was 0.89.

– Recall: The proportion of true positives among the total actual positives. The recall for healthy individuals was 0.77, and for sick individuals it was 0.86.

– F1-Score: The harmonic mean between precision and recall provides a balanced measure of the model's performance. The F1-Score for healthy individuals was 0.74, and for sick individuals it was 0.88.

– Support: The number of samples in each class, with 30 samples for healthy individuals and 66 for sick individuals.

In addition, aggregated metrics are presented to evaluate the overall performance of the model:

– Accuracy: The proportion of correct predictions out of the total predictions. The total accuracy of the model in the training set was 0.83, based on 96 samples.

– Macro Avg: The average of precision, recall, and F1-Score calculated in an unweighted manner across all classes, resulting in values of 0.80, 0.82, and 0.81 respectively.

– Weighted Avg: The average of precision, recall, and F1-Score weighted by the number of samples in each class, resulting in values of 0.84, 0.83, and 0.83 respectively.

Test Set Results The second section of the table presents the results of the model applied to a test dataset, specifically designed to evaluate its generalization capability. The following metrics are reported:

– Precision: The precision for the healthy individuals class in the test set was 0.71, and for sick individuals, it was 0.88.

– Recall: The recall for healthy individuals was 0.72, while for sick individuals, it was 0.87.

– F1-Score: The F1-Score for healthy individuals was 0.77, and for sick individuals it was 0.88.

– Support: The number of samples in each class was 10 for both healthy and sick individuals.

The aggregated metrics for the test set were as follows:

– Accuracy: The total accuracy of the model on the test set was 0.80, based on 20 samples.

– Macro Avg: The average of precision, recall, and F1-Score calculated in an unweighted manner across all classes, resulting in values of 0.80, 0.81, and 0.81 respectively.

– Weighted Avg: The average of precision, recall, and F1-Score weighted by the number of samples in each class, resulting in values of 0.81, 0.82, and 0.80 respectively.

11 Discussion of Results Pre-trained Model Results

The pre-trained model showed good performance in the classification of COPD. The results indicate that the precision for healthy individuals was 0.72, while for sick individuals it was 0.89. The model achieved an F1-score of 0.74 for healthy individuals and 0.88 for sick individuals, with support of 30 and 66 samples, respectively. The overall accuracy of the model was 0.83, with a macro average of 0.81 and a weighted average of 0.83. These results demonstrate the effectiveness of the pre-trained model in COPD classification, but they also suggest that the model could benefit from additional adjustments to improve precision and recall, especially in diverse populations. One of the primary limitations identified is the relatively small size of the dataset used.

Although the model has shown promising performance under controlled conditions, a larger and more diverse dataset is crucial for validating and enhancing its generalizability. a greater volume of data will allow the model to learn and adapt to a broader range of variations in vocal characteristics associated with COPD, thereby improving its accuracy and robustness. Additionally, validating the model in a realistic setting remains an outstanding issue. data obtained in a laboratory environment may not reflect the variable and potentially noisy conditions encountered in a real hospital setting. Environmental noise, medical equipment, and data capture devices could affect the quality of vocalization data, which may influence the accuracy of the model.

For example, background noise in a hospital or differences in the quality of hardware used for capturing vocalizations could introduce variations not observed under controlled laboratory conditions. The transformer model has been evaluated primarily in controlled environments, and more work is needed to adapt and validate it in real clinical situations. this includes conducting extensive tests in hospital settings and with data capture equipment that reflects practical use conditions. evaluating the model under these conditions will allow identification and addressing of possible biases or deficiencies, as well as making necessary adjustments to enhance its effectiveness in detecting COPD in diverse populations and clinical contexts

12 Study Results in Vulnerable Populations

When applying the model to specific data from vulnerable populations, the results showed a slight decrease in precision and F1-score. For healthy individuals in these populations, the precision was 0.71 and the F1-score was 0.77, with a support of 10 samples. For sick individuals, the precision was 0.88 and the F1-score was 0.88, also with a support of 10 samples. The overall accuracy in these populations was 0.80, with a macro average of 0.81 and a weighted average of 0.80. The observed decrease in precision and F1-score when applying the model to vulnerable populations indicates that the model, although robust, needs to be adjusted to account for specific factors in these populations, such as genetic and environmental characteristics. These results provide a solid foundation for future improvements to the model, to increase its precision and adaptability to different contexts and populations.

13 Conclusion

The study has demonstrated that the pre-trained model has a remarkable capacity to generalize to different datasets, which is crucial to avoid overfitting and ensure its applicability to diverse populations. This ability allows the model to maintain its effectiveness even when facing new and specific data from vulnerable populations, such as patients in the state of Guerrero. Despite some performance variations, the results indicate that the model possesses great flexibility and utility. Its ability to adapt to new data underscores the potential of pre-trained AI models for the classification of diseases such as COPD. The robustness of the model is evident in its consistency when comparing the original data with the new data, although a slight decrease in precision and F1-score was observed.

This decrease suggests specific areas for improvement, providing a solid foundation for future optimizations and adjustments. The differences observed in the model's sensitivity and specificity when analyzing patient data highlight the importance of incorporating specific genetic and environmental factors. These factors are essential for personalizing and improving the model, thus increasing its precision and applicability in different contexts and populations. The study's results offer valuable insights that can be used to fine-tune the model more precisely, enhancing its performance in diverse populations. Moreover, the study demonstrates that the pre-trained model can be a valuable tool in the classification of COPD, even in vulnerable populations.

This is especially relevant in contexts with resource limitations and specific population conditions, where a flexible and adaptable model can have a significant impact. It is important to note that these data are preliminary and that collaborations are currently underway with public and private hospitals to collect more data and with a population entirely different from the current one to diversify the model. this expansion of the database will allow for further evaluation and enhancement of the model's capacity to generalize to diverse populations, increasing its robustness and applicability in different clinical contexts.

This strategy will not only improve the model's precision and reliability but also contribute to its ability to adapt to real clinical scenarios, providing a valuable resource for the classification and management of COPD in various populations. Similarly, efforts will be made to validate the models with databases collected by other institutions. this validation will allow for the comparison and evaluation of the model's performance on varied and previously unseen datasets, ensuring that its application is truly universal and effective. External validation is essential for identifying potential biases and limitations of the model, providing opportunities for fine-tuning to improve its accuracy and clinical utility.

By integrating data from various sources, a more comprehensive and detailed understanding of COPD can be achieved, which in turn strengthens the model's ability to adapt to different epidemiological and socioeconomic realities, ensuring its effectiveness across a wide range of contexts and populations. In summary, the study's findings underscore the potential of pre-trained AI models for medical applications, demonstrating their effectiveness, flexibility, and ability to continuously improve through specific adjustments based on real and diverse data. These findings provide a solid foundation for future research and development, to optimize and adapt these models to the specific needs of different populations, thus improving precision and effectiveness in disease classification.

References

1. Alvarez-Casado, C., Lage-Cañellas, M., Pedone, M., Wu, X., Nguyen, L., Bordallo-López, M.: Respiratory disease classification and biometric analysis using biosignals from digital stethoscopes. In: Proceedings of the 32nd European Signal Processing Conference, pp. 1556–1560 (2024) doi: 10.23919/EUSIPCO63174.2024.10714964
2. Carbajal, D. S.: La contaminación intramuros del humo de biomasa. Universidad Y Sociedad, vol. 14, no. S1, pp. 396–402 (2022)

3. Castañeda-Torreblanca, M. J.: Diferencias clínicas, espirométricas y radiográficas en enfermedad pulmonar obstructiva crónica por biomasa y tabaco. Master's thesis, Universidad Privada Antenor Orrego (2022)
4. Chen, L. W., Rudnicky, A.: Exploring wav2vec 2.0 fine tuning for improved speech emotion recognition. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1–5 (2023) doi: 10.1109/ICASSP49357.2023.10095036
5. Cheng, W., Zhou, A., Zeng, Y., Lin, L., Song, Q., Liu, C., Zhou, Z., Peng, Y., Yang, M., Yang, L., Chen, Y., Cai, S., Chen, P.: Prediction of hospitalization and mortality in patients with chronic obstructive pulmonary disease with the new global initiative for chronic obstructive lung disease 2023 group classification: A prospective cohort and a retrospective analysis. International Journal of Chronic Obstructive Pulmonary Disease, vol. 18, pp. 2341–2352 (2023) doi: 10.2147/copd.s429104
6. Curioso, W. H., Brunette, M. J.: Inteligencia artificial e innovación para optimizar el proceso de diagnóstico de la tuberculosis. Revista Peruana de Medicina Experimental y Salud Pública, vol. 37, no. 3, pp. 554–8 (2020) doi: 10.17843/rpmesp.2020.373.5585
7. Elonheimo, H. M., Mattila, T., Andersen, H. R., Bocca, B., Ruggieri, F., Haverinen, E., Tolonen, H.: Environmental substances associated with chronic obstructive pulmonary disease—a scoping review. International Journal of Environmental Research and Public Health, vol. 19, no. 7, pp. 3945 (2022) doi: 10.3390/ijerph19073945
8. Expósito-Gallardo, M. C., Ávila-Ávila, R.: Aplicaciones de la inteligencia artificial en la medicina: Perspectivas y problemas. Acimed, vol. 17, no. 5 (2008)
9. Feng, Y., Wang, Y., Zeng, C., Mao, H.: Artificial intelligence and machine learning in chronic airway diseases: Focus on asthma and chronic obstructive pulmonary disease. International journal of medical sciences, vol. 18, no. 13, pp. 2871 (2021)
10. García-Frapolli, E., Schilmann, A., Berrueta, V. M., Riojas-Rodríguez, H., Edwards, R. D., Johnson, M., Guevara-Sanginés, A., Armendariz, C., Masera, O.: Beyond fuelwood savings: Valuing the economic benefits of introducing improved biomass cookstoves in the Purépecha region of Mexico. Ecological Economics, vol. 69, no. 12, pp. 2598–2605 (2010) doi: 10.1016/j.ecolecon.2010.08.004
11. González-Block, M. Á., Reyes-Morales, H., Cahuana-Hurtado, L., Balandrán, A., Méndez, E.: Mexico: Health system review. Health Systems in Transition, vol. 22, no. 2 (2020)
12. Hsu, W. N., Sriram, A., Baevski, A., Likhomanenko, T., Xu, Q., Pratap, V., Kahn, J., Lee, A., Collobert, R., Synnaeve, G., Auli, M.: Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. Interspeech, pp. 721–725 (2021) doi: 10.21437/interspeech.2021-236
13. Koul, A., Bawa, R. K., Kumar, Y.: Artificial intelligence techniques to predict the airway disorders illness: A systematic review. Archives of Computational Methods in Engineering, vol. 30, no. 2, pp. 831–864 (2022) doi: 10.1007/s11831-022-09818-4
14. Lovisotto, G., Finnie, N., Munoz, M., Murnmadi, C. K., Metzen, J. H.: Give me your attention: Dot-product attention considered harmful for adversarial patch robustness. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15213–15222 (2022) doi: 10.1109/CVPR52688.2022.01480
15. Mang, L. D.: Investigación y desarrollo de técnicas de procesado de señal e inteligencia artificial aplicadas a la recuperación de información biomédica a partir del análisis de señales sonoras respiratorias. Master's thesis, Universidad de Jaén (2024)
16. Novoselov, S., Lavrentyeva, G., Avdeeva, A., Volokhov, V., Gusev, A.: Robust speaker recognition with transformers using wav2vec 2.0 (2022) doi: 10.48550/ARXIV.2203.15095
17. Pérez-Alonso, D. A.: Análisis de señales de tos para detección temprana de enfermedades respiratorias. Master's thesis, Universidad de Valladolid, Escuela Técnica Superior de Ingenieros de Telecomunicación (2019)

18. Rani, S., Chaurasia, A., Dutta, M. K., Myska, V., Burget, R.: Machine learning approach for automatic lungs sound diagnosis from pulmonary signals. In: Proceedings of the 44th International Conference on Telecommunications and Signal Processing, pp. 366–371 (2021) doi: 10.1109/TSP52935.2021.9522663
19. Robertson, N. M., Centner, C. S., Siddharthan, T.: Integrating artificial intelligence in the diagnosis of COPD globally: A way forward. Chronic Obstructive Pulmonary Diseases: Journal of the COPD Foundation, vol. 11, no. 1, pp. 114–120 (2024) doi: 10.15326/jcopdf.2023.0449
20. Sierra-Villegas, S.: Caracterización de la enfermedad pulmonar obstructiva crónica (EPOC) a partir de imágenes de radiografías de tórax y datos clínicos. Master's thesis, Universidad Escuela de Ingeniería de Antioquia (2023)
21. Vargas-Ramírez, L., Ayazo, R. B.: Inteligencia artificial en neumología. Medicina, vol. 43, no. 4, pp. 570–581 (2021)

ISSN 1870-4069

SS-DTL: Semantic Segmentation with Dual Transfer Learning in Leukemic Retinopathy Using Knowledge from Diabetic Retinopathy

Edgar Gilberto Platas-Campero, Raquel Díaz-Hernández,
Leopoldo Altamirano-Robles

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Puebla, Mexico

{gilberto.platas, raqueld, robles}@inaoep.mx

Abstract. Leukemic retinopathy encompasses ocular pathologies affecting the retina, typically manifesting in leukemia's early stages. However, the scarcity of images limits the training of semantic segmentation models for lesion-level analysis. To address this, a Dual Transfer Learning (DTL) approach leverages the morphological similarities between lesions of diabetic and leukemic retinopathy, overcoming data limitations. Initially, Transfer Learning is applied in a U-Net network using ResNet-18, AlexNet, and VGG16 models trained with diabetic retinopathy images. The best-performing model, VGG16, achieves 95.34% accuracy and is selected for a second transfer stage. In this phase, the stored model is adapted to a modified U-Net architecture and retrained with leukemic retinopathy images. Results show that VGG16 attained a DICE coefficient of 95.16% and an IoU of 90.77% for diabetic retinopathy. For leukemic retinopathy, it achieved a DICE of 94.11% and an IoU of 88.88%. This demonstrates the effectiveness of DTL in addressing data scarcity, improving segmentation performance by transferring knowledge from better-studied diseases. The proposed method highlights the value of utilizing data from similar pathologies, representing significant progress in detecting leukemic retinopathy and advancing the study of lesser-known medical conditions.

Keywords: Semantic segmentation, dual transfer learning, diabetic retinopathy, leukemic retinopathy.

1 Introduction

Leukemia is a heterogeneous group of hematological neoplasms, arising from an abnormal increase of leukocytes in the bone marrow and peripheral blood [8]. Leukemic retinopathy (LR), on the other hand, is an ocular manifestation that occurs when leukemic cells infiltrate the retina, potentially extending to the optic nerve or optic disk [13, 19] his infiltration can cause various visual manifestations, such as hemorrhages, cotton-wool spots, Roth's spots, and exudates [33, 37]. Various case studies [26, 16, 41], have documented patients with leukemia developing these ocular manifestations.

Table 1. Comparison of SS and DTL related work.

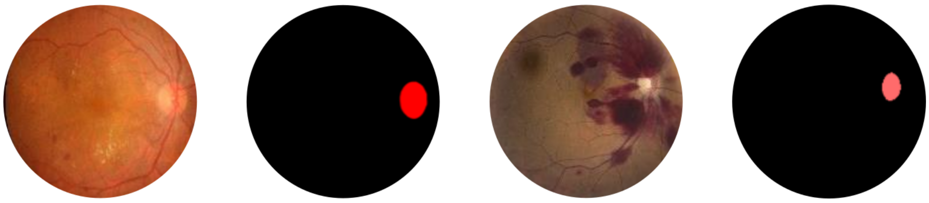
Author/year	Image	Disease	Category	SS	DTL
[34]/2024	O.D.	Glaucoma	Medical	X	-
[21]/2023	O.D. and O.C.	Glaucoma	Medical	X	-
[15]/2023	O.D. and O.C.	Glaucoma	Medical	X	-
[4]/2020	O.D. and O.C.	Glaucoma	Medical	X	-
[43]/2021	DDD17	-	Driving scene	X	X
[24]/2024	Brain MRI	Neurological	Medical	X	X
[6]/2022	COCO and ADE	-	Objects	X	X
[35]/2018	Plants	-	Vegetation	X	X
Proposed	**O.D.**	**D.R. and L.R.**	**Medical**	**X**	**X**

There is evidence [1, 11, 39, 3] that these manifestations can be the first visible signs of a possible leukemia, highlighting the importance of early ophthalmological detection in patients with non-specific symptoms. However, their accurate diagnosis presents significant challenges due to the need for high skill in differentiating types of retinopathies. In this context, deep learning presents itself as a solution. It is worth noting that, to date, only one study has been identified on detecting LR, focusing on disease classification [28].

This lack highlights an important area of opportunity, such as semantic segmentation, which could significantly improve the differentiation of this disease. Classical segmentation divides an image into distinct regions based on characteristics such as color, texture, or depth, grouping related pixels without assigning them to a specific class. In contrast, semantic segmentation [20, 14] assigns a specific label to each pixel, identifying classes such as “car,” “person,” or “tree,” providing a more detailed and specific understanding of the image. Model training for semantic segmentation requires images with segmented masks. However, no specific data for LR were found in public repositories, only for diabetic retinopathy (DR) [29, 9].

This lack of data hinders the training of deep learning models. In addition, the generation of synthetic images requires disease specific data, and the complex morphological characteristics of the lesions make this process even more difficult. As a solution, Dual Transfer Learning with DR databases is proposed. Transfer learning (TL) facilitates the adaptation of a model designed for one task to a second related task and is particularly effective when data are scarce or difficult to obtain [24, 42].

There are two approaches: feature extraction and fine-tuning. In feature extraction, all layers of a pretrained model are frozen, and a new classifier is added, ideal for large datasets. In fine-tuning, the last layers of a pretrained network are unfrozen, which is ideal for small datasets. The combination of both approaches is called Dual Transfer Learning (DTL). This methodology takes what is learned by one method and transfers it to the other [23]. In ophthalmology, specifically in DR, TL is adequate despite the complexity and variability of retinal images, providing positive results in the semantic segmentation of lesions in fundus images, as observed in [40, 44, 25, 36, 22].



(a) (b) (c) (d)

Fig. 1. Fundus images and true masks: a) DR, b) True mask DR, c) LR, d) True mask LR.

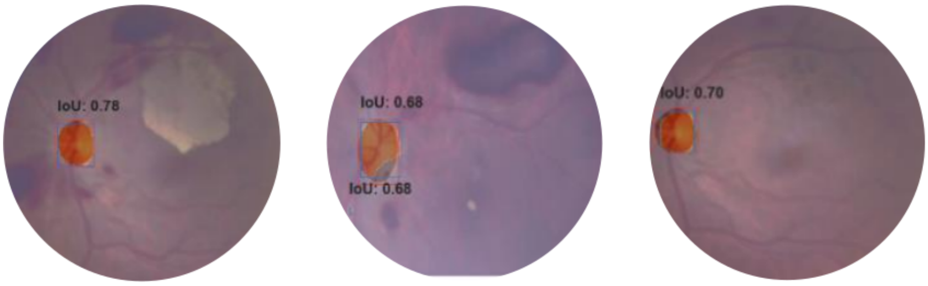
(a) (b)

Fig. 2. Fundus images: a) DR, b) LR.

Therefore, a DTL approach is proposed, starting with DR images and then applying it to leukemia retinopathy images. Before conducting the TL tests, the elements to be semantically segmented in the fundus images were selected, opting for a bi-class segmentation focused on the optic disc. The selection and identification of the optic disc is of paramount Semantic Segmentation with DTL importance in DR and related conditions, as it provides crucial information about the presence and progression of the disease [2, 17, 45, 27]. After selecting the data type, the first TL was applied using different CNN models (ResNet-18, AlexNet, VGG16), adapting them to a U-Net network. First, manually segmented LR images were used to observe the behavior with little data; subsequently, DR images were tested. The resulting model was stored.

Then, DTL was applied, employing the model previously trained with DR images, but this time training it with LR images. In the first case, we faced the problem of generating poor masks due to the limited data available for LR, so we applied the DTL strategy. This approach demonstrated better performance in generating masks associated with LR. The results obtained highlight the main objective of this study, which is to explore the effectiveness of the DTL technique in U-Net models for semantic segmentation of retinopathy images in data-constrained contexts. The main contributions of this paper are:

– Introduce a method using deep learning techniques for semantic segmentation of the optic disc in fundus images associated with LR.
– Propose a procedure to exploit the knowledge acquired from previously semantically segmented elements in DR images and adapt it to LR in scenarios where data are scarce or difficult to obtain.
– Present an alternative solution in data-scarce contexts where it is impossible to generate synthetic images of the condition due to the lack of data and the complexity of the lesions.



(a) (b) (c)

Fig. 3. Predicted optic disc masks for LR.

2 Related Works

In this section, the works related to this research are presented. Although there are numerous studies on classical segmentation [38, 7, 47, 46, 48], our focus is specifically on semantic segmentation and double TL techniques, as detailed below: The study in [34] proposes an ensemble semantic segmentation method to improve glaucoma detection using fundus images and deep learning models such as U-Net 3Plus, Deep Lab V3P, PSPNET, and UW-Net. On the other hand, [21] presents an encoder-decoder network for optic disc and optic cup segmentation in fundus images using a pretrained ConvNext encoder and a lightweight decoder enhanced with the Dual-Path Response Fusion Attention (DPRFA) module.

In addition, [15] introduces a modified U-Net model for optic disc and optic cup segmentation in fundus images, using a modified Atrous Spatial Pyramid Pooling (ASPP) module to capture detailed spatial data and improve segmentation accuracy. Finally, [4] proposes a deep learning-based approach for automatic segmentation of the optic disc and optic cup and a model for glaucoma detection in fundus images.

On the other hand, with regard to DTL related work, this study [43] suggests a DTL method for event-based final task prediction by converting events into detailed images containing more structural information. This information is transferred to the EEL branch, improving its ability to perform complex tasks. On the other hand, [24] discusses the unsupervised adaptation of a skull-stripping model on brain MRI images, which was trained with adult data to work with newborn images.

Also, [6] presents SimFormer, a method for semantic segmentation that reduces the need for detailed pixel-level labels. Based on MaskFormer, Sim-Former uses dual similarity transfer to learn new classes with image-level labels, transferring proposal-pixel similarities from known classes to new classes and learning inter-pixel similarities to improve segmentation. Finally, [35] explores the applicability of a fully convolutional network for plant segmentation using a two-step transfer method.

First, learning is transferred from a domain with many labeled data to a central category in the plant domain. Then, this major category is adapted to a minor category with little data. In Table 1, the reviewed articles are presented, detailing the author and year of publication under the “Author/Year” column; the type of medical image used, which includes Optic Disc (O.D.); Optic Cup (O.C.); DDD17 (DAVIS Driving Dataset 2017); COCO (Common Objects in Context); and ADE (ADE20K Dataset); the

Table 2. Comparison with other models using DR images.

Model	Accuracy	Loss	DICE	IoU
AlexNet	0.9082	0.0097	0.9122	0.8386
ResNet18	0.9416	0.0056	0.9520	0.9084
VGG16	**0.9560**	**0.0052**	**0.9581**	**0.9196**

disease studied; the application category; and the application of semantic segmentation (S.S.) and DTL techniques. The proposal of this research, listed at the end of Table 1, stands out for the joint application of semantic segmentation and Dual Transfer Learning (DTL) in diabetic retinopathy (DR) and leukemic retinopath (LR) images. This approach is particularly significant as, after the literature review, it is concluded that there is very little documentation on semantic segmentation using DTL in the medical field. Specifically, no previous studies have been found that address LR with these advanced techniques.

3 Characteristics of the Dataset Used

3.1 Dataset

The IDRID dataset [29], composed of segmented color fundus images, was used to perform this research. This set includes 81 images, divided into 54 for training and 27 for testing, all in JPG format. In addition, each image has its respective masks in TIF format. On the other hand, concerning the images of LR, an exhaustive review of medical publications was carried out to identify representative images of this condition. As a result of this search, 40 images were selected from various sources specialized in this disorder [12, 10, 18, 37, 32].

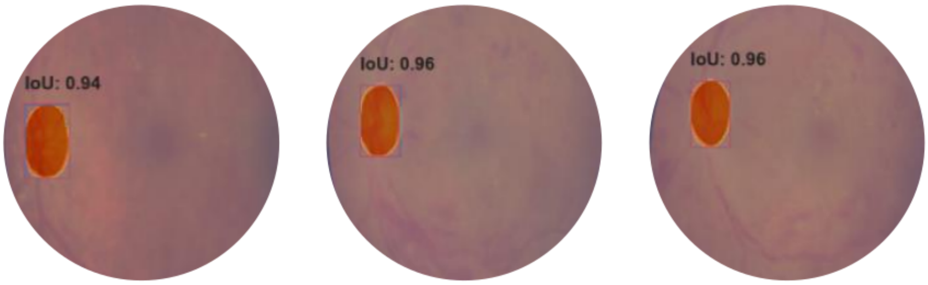
3.2 Dataset Distribution

The training set includes 54 optic discs for DR imaging. The test set includes 27 optic disc images. For the LR images, manual semantic segmentation of the optic disc was performed on each image using the freely licensed ImageJ program [30]. The dataset was organized into two groups: 32 images for training and 8 images for testing.

In both cases, the training and testing are done using the full fundus images in RGB format and the true masks of the optic disc corresponding to each case. The Fig. 1, shows the shape of the masks for each condition.

3.3 Comparison of the Optic Disc Between Retinopathies

The optic disc is a crucial structure in ophthalmologic evaluation, standing out as an oval area within the retina where blood vessels converge to exit into the eye. It generally presents a coloration that varies from pink to yellow; the temporal half is paler than the rest, and the nasal half has a less delineated border. These characteristics may be altered in the presence of certain diseases. Below are two images showing the presence of lesions and how they are associated with each retinopathy.



(a) (b) (c)

Fig. 4. Predicted optic disc masks for DR.

The Fig. 2, shows two fundus images of DR and LR, respectively. In the first image, we can see that the optic disc shows clarity in its well-defined borders; the disc coloration is normal, varying from pink to pale yellow, with no signs of pallor or hyperemia, suggesting good vascularization and absence of acute inflammation. In addition, the disc structure is intact, with no signs of hemorrhage or neovascularization.

On the other hand, in the second image, the optic disc shows several prominent pathologic features: the disc borders appear blurred, the disc shows an unusually pale coloration, especially in its temporal half, and hemorrhages are seen within the disc, indicative of possible vascular disorders or retinal damage. Although the optic discs in both retinopathies may appear different at first glance, crucial specific similarities justify using the transfer of learning approach. This approach allows leveraging previously acquired knowledge to improve diagnostic accuracy in less well-studied diseases.

4 Methodology Proposed

4.1 Proposed DTL Semantic Segmentation Model

This study proposes a methodology involving several tests with images of diabetic and LR to evaluate the effectiveness of TL. CNN models (ResNet18, AlexNet, VGG16) previously trained on ImageNet were selected and adapted to function as U-Net network's encoders. Subsequently, these adapted models were retrained and evaluated using data sets specific to each type of retinopathy. A plan was established that included three critical tests. The first aims to evaluate the effectiveness of TL using exclusively LR images to semantically segment the optic disc in fundus images, even with a limited number of images of this pathology.

In the second test, selected models are retrained by TL, applying a fine-tuning technique with DR images in a U-Net network. This process involves the partial freezing of the internal layers of the model and the adaptation of the final layers to the study's specific context. After this adjustment, the modified model is retained for further use. The third test focuses on LR using the model already adjusted for DR. A feature extraction strategy is implemented, freezing all the layers of the model, which are then adapted in the U-Net coding stage.

Table 3. Results obtained using DTL in LR images.

Model	Accuracy	Loss	DICE	IoU
VGG16	**0.9143**	**0.0090**	**0.9411**	**0.8888**

Table 4. Comparison of results for LR applying both techniques: TL vs DTL.

Image	Technique	Accuracy	Loss	DICE	IoU
Leukemia	TL	0.8351	0.0172	0.8765	0.7801
Leukemia	**DTL**	**0.9143**	**0.0090**	**0.9411**	**0.8888**

This model undergoes a new training process with the LR dataset, thus allowing the exploitation of previously learned features from the DR dataset to obtain a better semantic segmentation of the optic disc in LR images, even when the available data are limited.

4.2 Performance Metrics

Several metrics have been selected to evaluate the performance of the deep learning model in semantic segmentation tasks and provide different perspectives on its performance. Accuracy measures the proportion of correct predictions and gives an overview of the model's performance. The Loss function evaluates how much the model's predictions deviate from the actual values and is crucial for optimization during training. The DICE coefficient and the Jaccard Index (IoU) are specific metrics for segmentation quality assessment, measuring the similarity and overlap between model predictions and true annotations. The equation (1) presents the DICE coefficient [5]:

$$\mathrm{DICE} = \frac{2\,(X \cap Y)}{|X| + |Y|}, \tag{1}$$

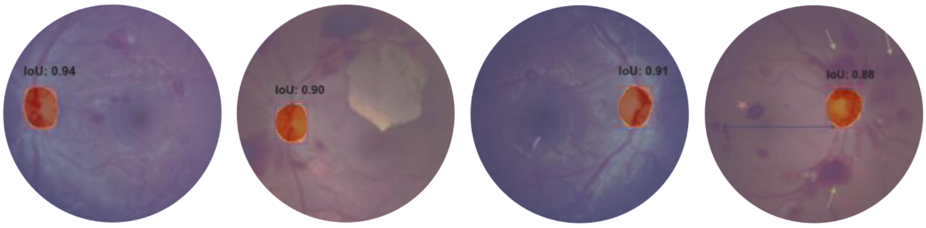
where, X represents the set of pixels predicted as positive (i.e., the segmentation performed by the model), and Y is the set of true positive pixels (the ground truth or real segmentation). On the other hand, the Jaccard Index [31], also known as Intersection over Union (IoU), is calculated by the equation (2):

$$\mathrm{IoU} = \frac{|X \cap Y|}{|X \cup Y|}. \tag{2}$$

In this metric, X is the model prediction, and Y represents the ground truth. The IoU evaluates the effectiveness of the segmentation by dividing the size of the intersection of the sets X and Y by the size of their union, providing a clear indicator of the accuracy of the segmentation.

5 Results

In the first test, which was trained exclusively with images of LR, the U-Net architecture was used with a pretrained VGG16 model. The results obtained for the first test, including the predicted optic disc masks and the calculated Index of Overlap (IoU), are presented in Fig. 3. In the second test, DR images were used, and TL was applied



(a) (b) (c) (d)

Fig. 5. Optical disc predicted masks with DTL in LR images.

with a fine-tuning approach using various pretrained models adapted to the U-Net architecture. The results obtained are presented in Table 2, and the predicted mask along with the calculated Index of Overlap (IoU) is shown in Fig. 4. In the third test, a DTL approach focused on feature extraction was implemented by adapting the U-Net network with the model previously trained with DR data. Subsequently, this model was retrained using LR images, taking advantage of the knowledge acquired in the context of DR. The outcomes are displayed in Table 3, while Fig. 5. exhibits the predicted mask along with the calculated Index of Overlap (IoU). As shown in Table 4, DTL showed better performance compared to TL by leveraging prior knowledge from DR. This behavior indicates that DTL is more effective at transferring knowledge between similar domains, significantly improving the results in the segmentation of LR images.

6 Discussion

As could be seen in Fig. 3, which was trained only with LR images, the evaluated model fails to fully satisfy the requirements of accurate semantic segmentation for the optic disc. The predicted masks do not cover the entire optic disc in cases a), b), and c), reaching overlap indices (IoU) of 0.78, 0.68, and 0.70, respectively. This result indicates that the match between the predicted mask and the actual region of the optic disc is insufficient, especially when the model is trained exclusively with images of LR. In the second test, as seen in Table 2, the VGG16 model proved superior to the others, achieving an (Accuracy) of 0.9560, notably higher than the 0.9416 of ResNet18 and the 0.9082 of AlexNet. In addition, VGG16 achieved the highest values in the DICE (0.9581) and IoU (0.9196) metrics, highlighting its efficiency in optic disc specific semantic segmentation.

The results of the predicted masks, shown in Fig. 4, with Index of Overlap (IoU) values of 0.94 for a), 0.96 for b), and 0.96 for c), confirm that VGG16 offers a significant improvement in terms of semantic segmentation accuracy and quality compared to other evaluated models. In the third test, which employed a DTL approach using LR images, the VGG16 model achieved remarkable results, as shown in Table 3. The Accuracy obtained was 0.9143, while the value of the Loss function was recorded at 0.0090. Additionally, the model demonstrated high performance in the semantic segmentation evaluation metrics, with a DICE coefficient of 0.9411 and an Index of Overlap (IoU) of 0.8888. On the other hand, Fig. 5, presents the predicted masks for the optic disc using the DTL approach on DR images.

The Index of Overlap (IoU) values for each image are as follows: for image a) an IoU of 0.94 was achieved, image b) reached an IoU of 0.90, c) repeated an IoU of 0.91, and finally, image d) recorded an IoU of 0.88. Finally, a comparison was made applying both techniques with the LR dataset. The Table 4 shows that the DTL technique significantly outperforms TL in all metrics (accuracy, loss, DICE, IoU) for leukemia retinopathy images.

These results illustrate the variability in semantic segmentation accuracy and highlight the positive impact of using advanced techniques such as DTL to improve optic disc identification and delineation in complex pathological conditions. In addition, they emphasize the advantages of applying this technique in situations where little information is available, demonstrating its usefulness in managing diagnostic challenges such as those presented in this research.

7 Conclusion

This research addressed the problem of missing data and the difficulty in generating synthetic images due to the morphological characteristics of lesions and the scarcity of data in LR. In these scenarios, DTL enabled leveraging prior knowledge acquired in one specific domain to improve performance in another significantly. This approach optimized the use of limited data. It improved the results obtained in the semantic segmentation model, proving an effective solution to data limitation and difficulty in generating synthetic images.

8 Future Works

Future steps of this research involve expanding semantic segmentation to other lesions characteristic of diabetic and LR, such as microaneurysms, hemorrhages, exudates, Rorh's spots, and leukemic infiltrates. These elements present additional challenges due to their greater variability and lower frequency than the optic disc.

Acknowledgments. We would like to thank the Consejo Nacional de Humanidades, Ciencias y Tecnologías (Conhacyt) for the financial support through the doctoral scholarship.

References

1. Agarwal, M.: Bilateral proliferative retinopathy with blood-tinged hypopyon as the first manifestation of chronic myeloid leukemia. Indian Journal of Ophthalmology - Case Reports, vol. 3, no. 2, pp. 638 (2023) doi: 10.4103/ijo.ijo_150_23
2. Alam, M., Zhao, E., Lam, C., Rubin, D.: Segmentation-assisted fully convolutional neural network enhances deep learning performance to identify proliferative diabetic retinopathy. Journal of Clinical Medicine, vol. 12, no. 1, pp. 385 (2023) doi: 10.3390/jcm12010385

3. Alamri, S. S., Jammah, A. A., AlKhaldi, S. A., AlKharashi, A. A., Aljatham, A. A.: Optic disc infiltration as a first sign of acute lymphoblastic leukemia relapse: A case report. The Open Ophthalmology Journal, vol. 18, no. 1 (2024) doi: 10.2174/0118743641302926240520102811
4. Božić-Štulić, D., Braović, M., Stipaničev, D.: Deep learning based approach for optic disc and optic cup semantic segmentation for glaucoma analysis in retinal fundus images. International journal of electrical and computer engineering systems, vol. 11, no. 2, pp. 111–120 (2020) doi: 10.32985/ijeces.11.2.6
5. Carass, A., Roy, S., Gherman, A., Reinhold, J. C., Jesson, A., Arbel, T., Maier, O., Handels, H., Ghafoorian, M., Platel, B., Birenbaum, A., Greenspan, H., Pham, D. L., Crainiceanu, C. M., Calabresi, P. A., Prince, J. L., Roncal, W. R. G., Shinohara, R. T., Oguz, I.: Evaluating white matter lesion segmentations with refined sørensen-dice analysis. Scientific Reports, vol. 10, no. 1 (2020) doi: 10.1038/s41598-020-64803-w
6. Chen, J., Niu, L., Zhou, S., Si, J., Qian, C., Zhang, L.: Weak-shot semantic segmentation via dual similarity transfer (2022) doi: 10.48550/ARXIV.2210.02270
7. Chen, Y., Bai, Y., Zhang, Y.: Optic disc and cup segmentation for glaucoma detection using attention U-Net incorporating residual mechanism. PeerJ Computer Science, vol. 10, pp. e1941 (2024) doi: 10.7717/peerj-cs.1941
8. Chennamadhavuni, A., Lyengar, V., Mukkamalla, S. K. R., Shimanovsky, A.: Leukemia. In: StatPearls, StatPearls Publishing (2023)
9. Decencière, E., Cazuguel, G., Zhang, X., Thibault, G., Klein, J. C., Meyer, F., Marcotegui, B., Quellec, G., Lamard, M., Danno, R., Elie, D., Massin, P., Viktor, Z., Erginay, A., Laÿ, B., Chabouis, A.: Teleophta: Machine learning and image processing methods for teleophthalmology. IRBM, vol. 34, no. 2, pp. 196–203 (2013) doi: 10.1016/j.irbm.2013.01.010
10. Gim, Y., Kim, H. J.: Ocular symptom can be the first presentation of differentiation syndrome in acute promyelocytic leukemia. Korean Journal of Ophthalmology, vol. 35, no. 1, pp. 94–96 (2021) doi: 10.3341/kjo.2020.0101
11. Han-Menz, C., Kozgar, S. A. M., Rana, M. S.: Periorbital cellulitis as the initial presentation of acute lymphoblastic leukaemia. BMJ Case Reports, vol. 17, no. 5, pp. e259087 (2024) doi: 10.1136/bcr-2023-259087
12. Istrate, M., Ciubotaru, A., Hasbei-Popa, M., Boariu, A. M., Iliescu, D. A.: Leukemic retinopathy, the first expression in a case of chronic myelomonocytic leukemia - a case report. romanian journal of ophthalmology, vol. 64, no. 4, pp. 419–422 (2020) doi: 10.22336/rjo.2020.65
13. Johnson, G. M., Rossen, J. L., Simon, S. S., Weinstein, J. L., Ralay-Ranaivo, H., Yoon, H. H.: Leukemic optic neuropathy in pediatric patients: A case series. Journal of Pediatric Ophthalmology and Strabismus, vol. 61, no. 1, pp. 67–72 (2024) doi: 10.3928/01913913-20230331-03
14. Karimi, D., Warfield, S. K., Gholipour, A.: Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations. Artificial Intelligence in Medicine, vol. 116, pp. 102078 (2021) doi: 10.1016/j.artmed.2021.102078
15. Kedari, B., Kamath, R., Arra, A., G, S., S, G.: Semantic segmentation of optic disc and optic cup using deep learning. In: 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–6 (2023) doi: 10.1109/icccnt56998.2023.10308314
16. Kochhar, D., Kaushik, J., Singh, A., Saraswat, A., Parihar, J. K. S.: Ocular manifestations of acute myeloid leukemia during induction phase of chemotherapy: A case series. Latin American Journal of Ophthalmology, vol. 7, pp. 1 (2024) doi: 10.25259/lajo_20_2023

17. Li, X., Xie, J., Zhang, L., Cui, Y., Zhang, G., Wang, J., Zhang, A., Chen, X., Huang, T., Meng, Q.: Differential distribution of manifest lesions in diabetic retinopathy by fundus fluorescein angiography and fundus photography. BMC Ophthalmology, vol. 20, no. 1 (2020) doi: 10.1186/s12886-020-01740-2
18. Lim, J., Kim, I., Sagong, M.: Leukostasis retinopathy as the first sign of chronic myeloid leukemia with severe hyperleukocytosis: Two case reports. Korean Journal of Ophthalmology, vol. 37, no. 3, pp. 266–269 (2023) doi: 10.3341/kjo.2023.0033
19. Liu, M., Chen, S., Zhang, D.: Learning attribute relation in attribute-based zero-shot classification. In: Intelligent Science and Intelligent Data Engineering. Lecture Notes in Computer Science, vol. 7751, pp. 514–521 (2013) doi: 10.1007/978-3-642-36669-7_63
20. Liu, X., Deng, Z., Yang, Y.: Recent progress in semantic image segmentation. Artificial Intelligence Review, vol. 52, no. 2, pp. 1089–1106 (2018) doi: 10.1007/s10462-018-9641-3
21. Mallick, S., Paul, J., Sil, J.: Response fusion attention U-ConvNext for accurate segmentation of optic disc and optic cup. Neurocomputing, vol. 559, pp. 126798 (2023) doi: 10.1016/j.neucom.2023.126798
22. Moura, C., Cortez, P., Assis, D., Motta, P., Silva, B.: Yolov8 deep learning model for diabetic retinopathy fundus image segmentation and disease classification. In: Anais do XVI Congresso Brasileiro de Inteligência Computacional, pp. 1–7 (2024) doi: 10.21528/cbic2023-159
23. Mukhlif, A. A., Al-Khateeb, B., Mohammed, M. A.: Incorporating a novel dual transfer learning approach for medical images. Sensors, vol. 23, no. 2, pp. 570 (2023) doi: 10.3390/s23020570
24. Omidi, A., Shamaei, A., Verschuu, A., King, R., Leijser, L., Souza, R.: Unsupervised domain adaptation of brain MRI skull stripping trained on adult data to newborns: Combining synthetic data with domain invariant features. In: Medical Imaging with Deep Learning, pp. 1–13 (2024)
25. Padmasini., N., Krithika., G., Lithiga., P., Akshaya., S.: Automatic detection and segmentation of retinal manifestations due to diabetic retinopathy. In: Proceedings of the International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication, pp. 1–6 (2023) doi: 10.1109/iconscept57958.2023.10170621
26. Padvi, U. I., Solu, T. M.: Ocular manifestations of leukaemia: A cross-sectional study from tertiary care hospital of South Gujarat, India. Journal of Clinical and Diagnostic Research, vol. 18, no. 4, pp. NC01–NC04 (2024) doi: 10.7860/jcdr/2024/69407.19293
27. Panigrahi, P., Thareja, J., Mishra, S., Minj, A.: Unilateral optic nerve leukemic infiltration due to relapse in a case of acute lymphoblastic leukaemia. TNOA Journal of Ophthalmic Science and Research, vol. 61, no. 2, pp. 231 (2023) doi: 10.4103/tjosr.tjosr_59_22
28. Platas-Campero, E. G., Díaz-Hernández, R., Altamirano-Robles, L.: Recognition of leukemic retinopathy using knowledge of diabetic retinopathy. In: Proceedings of Mexican Conference on Pattern Recognition Xalapa, Mexico, pp. 243–252 (2024) doi: 10.1007/978-3-031-62836-8_23
29. Porwal, P., Pachade, S., Kamble, R., Kokare, M., Deshmukh, G., Vivek, S., Fabrice, M.: Indian diabetic retinopathy image dataset (idrid) (2018) doi: 10.21227/H25W98
30. Rasband, W.: Imagej - image processing and analysis in java. Astrophysics Source Code Library, vol. 1, pp. 6013 (2012)
31. Real, R., Vargas, J. M.: The probabilistic basis of Jaccard's index of similarity. Systematic Biology, vol. 45, no. 3, pp. 380–385 (1996) doi: 10.1093/sysbio/45.3.380
32. Reddy, S. C., Jackson, N.: Retinopathy in acute leukaemia at initial diagnosis: Correlation of fundus lesions and haematological parameters. Acta Ophthalmologica Scandinavica, vol. 82, no. 1, pp. 81–85 (2003) doi: 10.1046/j.1600-0420.2003.00197.x

33. Rossi, C., Buizza, A., Alessio, G., Borselli, M., Taloni, A., Carnevali, A., Carnovale-Scalzo, G., Lucisano, A., Scorcia, V., Giannaccare, G.: Ophthalmic manifestations in patients with blood malignancies. Hematology Reports, vol. 16, no. 2, pp. 193–203 (2024) doi: 10.3390/hematolrep16020020
34. Sahab, Y.: Improving glaucoma detection: Harnessing the power of ensemble semantic segmentation for optic disc and optic cup with deep learning. Aswan University Journal of Sciences and Technology, vol. 4, no. 1, pp. 1–14 (2024) doi: 10.21608/aujst.2024.337875
35. Sakurai, S., Uchiyama, H., Shimada, A., Arita, D., Taniguchi, R. I.: Two-step transfer learning for semantic plant segmentation. In: Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods, pp. 332–339 (2018) doi: 10.5220/0006576303320339
36. Sathiyaseelan, R., Ranganathan, K., Ramamoorthy, R., Pedda Chennaiah, M.: Haemorrhage diagnosis in colour fundus images using a fast-convolutional neural network based on a modified U-Net. Network: Computation in Neural Systems, pp. 1–22 (2024) doi: 10.1080/0954898x.2024.2310687
37. Sayadi, J., Gouider, D., Allouche, Y., Choura, R., Cherni, I., Sayadi, M., Benneji, H., Zghal, I., Malek, I., Nacef, L.: Ophthalmic manifestations of newly diagnosed acute leukemia patients in a tunisian cohort. Clinical Ophthalmology, vol. 16, pp. 3425–3435 (2022) doi: 10.2147/opth.s365648
38. Septiarini, A., Hamdani, H., Setyaningsih, E., Junirianto, E., Utaminingrum, F.: Automatic method for optic disc segmentation using deep learning on retinal fundus images. Healthcare Informatics Research, vol. 29, no. 2, pp. 145–151 (2023) doi: 10.4258/hir.2023.29.2.145
39. Shateri-Amiri, B., Aliabadi, F., Ghalamkari, M., Omidali, M., Babaheidarian, P., Ayoubpour, M., Hejrati, A., Bubur-Jan-Bozorgi, N., Rekabi, V.: Acute myeloid leukemia first presentation with central nervous system (CNS) involvement: A case report. Discover Medicine, vol. 1, no. 1 (2024) doi: 10.1007/s44337-024-00010-0
40. Shaukat, N., Amin, J., Sharif, M., Azam, F., Kadry, S., Krishnamoorthy, S.: Three-dimensional semantic segmentation of diabetic retinopathy lesions and grading using transfer learning. Journal of Personalized Medicine, vol. 12, no. 9, pp. 1454 (2022) doi: 10.3390/jpm12091454
41. Thareja, J., Minj, A., Samal, P., Panigrahi, P. K.: Ophthalmic manifestations in hematological malignancies: An observational study from a tertiary care health center in eastern India. Indian Journal of Ophthalmology, vol. 72, no. 5, pp. 659–663 (2024) doi: 10.4103/ijo.ijo_815_23
42. Vrbancic, G., Podgorelec, V.: Transfer learning with adaptive fine-tuning. IEEE Access, vol. 8, pp. 196197–196211 (2020) doi: 10.1109/access.2020.3034343
43. Wang, L., Chae, Y., Yoon, K. J.: Dual transfer learning for event-based end-task prediction via pluggable event to image translation. In: International Conference on Computer Vision, pp. 2135–2145 (2021) doi: 10.48550/arXiv.2109.01801
44. Wang, S., Chen, Y., Yi, Z.: nmODE-Unet: A novel network for semantic segmentation of medical images. Applied Sciences, vol. 14, no. 1, pp. 411 (2024) doi: 10.3390/app14010411
45. Wang, Y., Zhu, L., Wang, H., Li, Z., Bai, R., Wei, Q., Huang, L., Xu, Y., Li, S., Chen, H.: Bilateral optic nerve infiltration and leukemic retinopathy as initial signs of leukemia relapse with central nervous system involvement in an adult: A case report. BMC Ophthalmology, vol. 24, no. 1 (2024) doi: 10.1186/s12886-024-03486-7
46. Yi, Y., Jiang, Y., Zhou, B., Zhang, N., Dai, J., Huang, X., Zeng, Q., Zhou, W.: C2FTFNet: Coarse-to-fine transformer network for joint optic disc and cup segmentation. Computers in Biology and Medicine, Elsevier BV, vol. 164, pp. 107215 (2023) doi: 10.1016/j.compbiomed.2023.107215

47. Zhang, J., Mei, C., Li, Z., Ying, J., Zheng, Q., Yi, Q., Wang, L.: Automated segmentation of optic disc and cup depicted on color fundus images using a distance-guided deep learning strategy. Biomedical Signal Processing and Control, vol. 86, pp. 105163 (2023) doi: 10.1016/j.bspc.2023.105163
48. Zhou, Z., Zheng, Y., Zhou, X., Yu, J., Rong, S.: Self-supervised pre-training for joint optic disc and cup segmentation via attention-aware network. BMC Ophthalmology, vol. 24, no. 1 (2024) doi: 10.1186/s12886-024-03376-y

ISSN 1870-4069

Role of Sparse Training and Evolutionary Optimization in Volatility Forecasting Models

Juan Francisco Muñoz-Elguezabal, Diego F. Arriaza-Alonzo

Instituto Tecnológico y de Estudios Superiores de Occidente,
Departamento de Matemáticas y Física, Tlaquepaque,
Mexico

{francisco@iteso.mx, diego.arriaza}@iteso.mx

Abstract. ETH is the native cryptocurrency of the Ethereum network, renowned for its smart contracts and diverse decentralized ecosystem. This research addresses the challenge of short-term volatility forecasting of ETH/USDT on a 10-minute interval, leveraging order book data and public trade data from the previous 30 minutes. Order book data includes buy and sell orders over time, while public trades refer to executed orders. Features derived from these data sources are used as model predictors. The first experiment tested the average past volatility, GARCH(1,1), LSTM, and an Auto-Encoder using a single training, validation, and test set. The second experiment applied the same models within a Walk-Forward architecture. The third experiment utilized a Time Fold Sequential Validation (T-Folds SV) technique, creating 10 folds and omitting 50 minutes between training and validation sets to prevent leakage. By calculating Kullback-Leibler Divergence, 5 folds were selected that have the characteristics of being different from each other and provide unique information. As a consequence, RAM consumption was significantly reduced while maintaining comparable results to previous experiments. Hyperparameter optimization with less data is now possible and is performed by Genetic Algorithms. After three generations of 750 models for both LSTM and Auto-Encoder, the best hyperparameter values were found, with the optimized LSTM model outperforming its counterpart. An ablation study as the last experiment was analyzed by removing the early stopping criteria of the best models, resulting in worse performance, but not significantly.

Keywords: Ether, order book, public trades, LSTM, auto-encoder, T-Folds SV, Kullback-Leibler divergence, genetic algorithms.

1 Introduction

Ethereum is a decentralized blockchain that was first introduced in 2013 by Vitalik Buterin, a computer programmer and researcher in cryptocurrency. It is a network that acts as the foundation for applications, organizations, communities and digital assets that anyone can create and utilize. Its main feature is its smart contract functionality, which is also the main difference between Bitcoin and Ethereum.

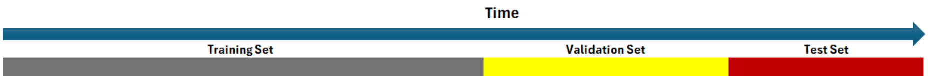


Fig. 1. One window training-validation-test setup.

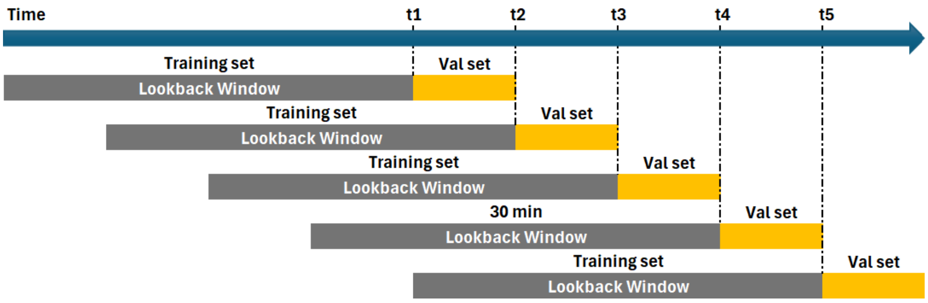


Fig. 2. Walk forward setup.

Smart contracts are computer programs that are executed when triggered by a transaction and act as building blocks for decentralized applications and organizations. Once a smart contract is published, it will be online and operational until Ethereum no longer exists. Examples of smart contracts are lending applications, insurance, decentralized trading exchanges, NFTs, etc. Ether is the native cryptocurrency of Ethereum. For ETH payments or use of Ethereum applications, a fee in ETH is charged, which is an incentive for a block producer to perform processing and verifications. Its main characteristics are that it is secured by cryptography, there is no intermediary service to make payments (it is peer-to-peer), there is no institution that can decide to print more ETH, or change the terms of use and it is divisible up to 18 decimal places, so it is not mandatory to buy 1 whole ETH.

ETH can also be exchanged for other currencies, products and services. The exchange of ETH with currencies is done on exchange services, where buy and sell orders are stored on the order book. Buy or bid orders represent an intention to buy a certain amount of ETH at some specified price while sell or ask orders represent the opposite. The exchange is done by matching orders by price from the order book into a trade transaction between buyers and sellers. Orders are one of the key components to understand the intention of the market as well as their influence on volatility. The cryptocurrency has gained relevance over the years and today is the second cryptocurrency with the highest market cap.

Due to its importance on the cryptocurrency market and its role on the Ethereum network, where at the end of year 2023 there are 96M accounts with ETH, 53.3M smart contracts, 410B value secured and 4k projects built, understanding and modeling ETH volatility is crucial for risk management and decision making. Therefore, the purpose of this work is to predict the short-term n-minutes volatility of ETH/USDT from the Binance Exchange with Order Book and Public Trades high frequency data. This work has no intention to produce financial models of volatility or the order book dynamics itself. The main contributions are to demonstrate the effectiveness and relevance of three crucial characteristics for Volatility Forecasting:

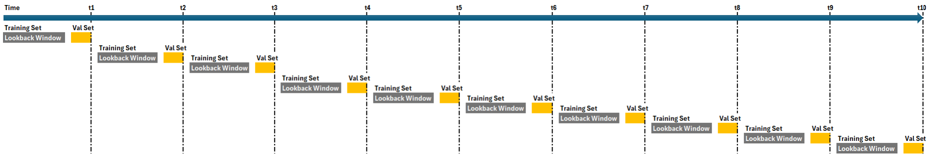


Fig. 3. T-folds SV.

The use of Order Books and Public Trades as source of information, the sparse training using T-Folds SV and KL Divergence for LSTM and Auto-Encoder models and Genetic Algorithms for Optimization method. The remainder of the paper is structured as follows. In section 2 an overview of the related work is presented. In section 3 information on how the data was collected and structured is provided as well as how the features of Order Book and Public Trades were calculated.

Theoretical explanation on how the models are formulated and their main properties is provided and evaluations metrics are also approached. In section 4details on how the models were setup in different experiments are given. Finally in section 5 and section 6 the results of the experiments are reported and discussed and conclusions on their performance are detailed.

2 Background and Related Work

The volatility estimation is a topic that has been extensively studied and developed over the years. In the specific case of cryptocurrencies, papers have focused on modeling BTC volatility using GARCH models. Vivian Naimy and Marianne Hayek compared the predictive ability of three GARCH models: GARCH (1,1), EWMA, and EGARCH (1,1). The results indicate that the asymmetric EGARCH (1,1) model outperforms the symmetric GARCH (1,1) and EWMA models in both in-sample and out-of-sample contexts. This suggests that BTC behavior differs from traditional currencies [9].

This paper, however, will only focus on GARCH to have a baseline model and it will be compared to only averaging the past volatility to measure its robustness. BTC will not be examined because it is the preferred cryptocurrency for analysis and not many papers focus on others such as Ether. On the survey made by Charandabi and Kamyar [2], they highlighted two papers.

The first one is the work of Miura, Pichl and Kaizoji called Artificial Neural Networks for Realized Volatility Prediction in Cryptocurrency Time Series [7], in which they aggregated RV values using 1-minute-sampled Bitcoin returns over 3-h intervals to predict future values using ANN (MLP, GRU, LSTM), SVM, and Ridge Regression and compare their results with Heterogeneous Auto-Regressive Realized Volatility (HARRV) model with optimized lag parameters.

Ridge Regression was able to outperform all the models. The other highlighted work was from Jang and Lee, in which they compared the Bayesian neural network with benchmark models, such as Linear Regression and SVR, on modeling and predicting the Bitcoin pricing process and concluded that the BNN outperformed the others. [4] Charandabi and Kaymer also pointed out the need for further research in areas like implementing data from less volatile cryptocurrencies and exploring new models.

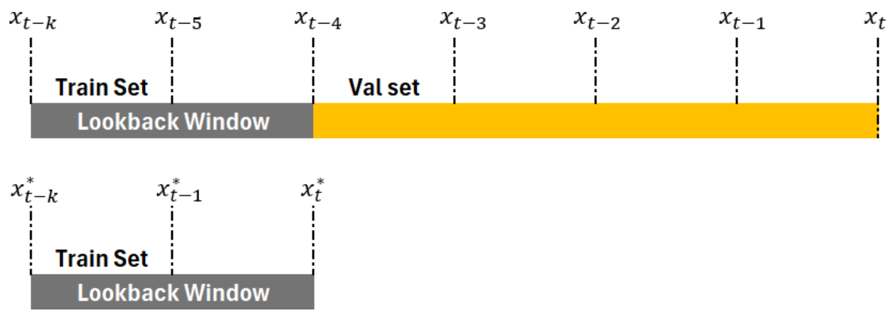


Fig. 4. Information leakage. X_{t-4}, X_{t-5} and X_{t-k} are input variables in the validation set, but they are the same inputs in the training set represented as X^*_t, X^*_{t-1} and X^*_{t-k} respectively.

As stated above, there have been papers that have worked with Deep Learning Models to forecast Time Series and Volatility. German Rodikov and Nino Antulov-Fantulin approached the challenge of predicting volatility by trying the LSTM model effectiveness against EWMA, HAR-RV, ARIMA, GARCH and GJR-GARCH. According to their results, LSTM outperformed all models in a rolling window between 5 and 12 periods. [10]. This work will also approach to work with the LSTM model, so it can be used to forecast 10 periods, specifically 10 minutes and it will also be compared to the baseline model GARCH.

Jung and Choi in their work called Forecasting Foreign Exchange Volatility Using Deep Learning Autoencoder-LSTM Techniques [5] predicted the FX volatility with a hybrid model that combined long short-term memory (LSTM) and autoencoder models, in which an LSTM model is utilized as an encoder and decoder inside an autoencoder network. They compared this hybrid model with the traditional LSTM model and based on their empirical results, the hybrid model outperformed the LSTM model. This work will also run an Auto-Encoder model similar to work by Jung and Choi because it will use LSTM layers on both the encoder and decoder and will also compare the Auto-Encoder model with the LSTM to see if it also outperforms it or not.

Guo, Bifet and Antulov-Fantulin point out that Order Book information cryptocurrency forecasting is still under-researched. Therefore, their proposal was to implement a temporal mixture model capable of adaptively exploit both volatility history and order book features. To forecast the volatility, they used a rolling strategy and divided the range of data into non-overlapping intervals, with each interval corresponding to one month. This paper is also contributing to forecast cryptocurrency volatility using Order Book information by modeling GARCH and LSTM. For training and validation Guo, Bifet and Antulov-Fantulin used two rolling strategies. The first one consists of training two months and validate with the immediate next month.

Then the training set moves to the next month as well as the validation set and the current training set is now taking in consideration the month used by the previous validation set for training. The second is similar, but instead of rolling both training set and validation set one month, the training set increase its periods by one month and only the validation set moves to next month.[3]. Regarding the training and validation of this paper, a Walk-Forward architecture will be applied as well as an approach similar to rolling strategy used by the three previous mentioned papers.

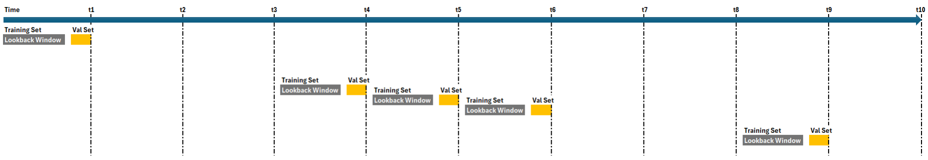


Fig. 5. KL Divergence T-fold SV.

The main difference of the last mentioned approach is that training sets will never be retrained on previous validation sets and there will be a purge of the observations of the training set that overlap with the validation set to avoid information leakage, which is highlighted by Marco Lopez de Prado[6] on his book Advances in Financial Machine Learning and is applied by Juan Francisco Muñoz and Juan Diego Sanchez on their poster T-Fold Sequential Validation Technique for Out-Of-Distribution Generalization with Financial Time Series Data[8].

3 Data and Methods

In this section, it is described how the data was collected and its source of information as well as calculations of Order Book and Public Trades features, Return and Volatility. Lastly, a theoretical framework addresses the Kullback-Leibler Divergence and the method used to optimize the hyperparameters, which is the Genetic Algorithm process.

3.1 Data Collection

There are two sources of information: Order Book Snapshots and Public Trades. Both were obtained by the website Tardis.dev. For every day, there were about 750-700K samples for Order Book data and around 500K samples for Public Trades data and both were from the exchange Binance only. The frequency of order books data was of 100 milliseconds and microseconds for Public Trades. The raw Order Book data provided the top 25 asks and bid prices as well as the top 25 asks and bids amounts. As for the raw Public Trades data, it displayed the trade price, the trade amount and the liquidity taker, which were only two possible values: buy or sell.

3.2 Order Books

The Order Book is a list of buy and sell orders that are waiting to be traded. All the features of an Order Book can be calculated on different levels (depth of orders) and can be uses as input variables for volatility modeling. As an example, it is possible to calculate the Volume Weighted Average Price (VWAP) of the first 5 levels, the first 10 levels or only the first level called Top of the Book (TOB). These are some of the following features equations:

$$\mathrm{VWAP} = \frac{\sum_{i=1}^{n} \text{bid price}_i \times \text{bid volume}_i + \sum_{i=1}^{n} \text{ask price}_i \times \text{ask volume}_i}{\sum_{i=1}^{n} \text{bid volume}_i + \sum_{i=1}^{n} \text{ask volume}_i}, \quad (1)$$

Table 1. Hyperparameter values.

Hyperparameters	**Values**
Epochs	[50,100,200]
Learning Rate	[0.0001, 0.001, 0.01, 0.05, 0.1]
Batch Sizes	[32, 64, 128, 256, 512]
Dropout	[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]
Optimizer	[RMSprop, SGD, Adam, Nadam, Adadelta]
Loss function	[MAE, MSE, HUBER, LOGCOSH]
Layers	[1-3]
Unit in Layers	[50, 100, 200]

$$\text{Total Depth} = \sum_{i=1}^{n} \left(\text{bid volume}_i + \text{ask volume}_i\right), \tag{2}$$

$$\text{Imbalance} = \sum_{i=1}^{n} \frac{\text{bid volume}_i}{\text{bid volume}_i + \text{ask volume}_i}, \tag{3}$$

$$\text{Total Spread} = \sum_{i=1}^{n} \frac{\text{Depth}_i}{\text{Total Depth}} \times \left(\text{price ask}_i - \text{price bid}_i\right), \tag{4}$$

$$\text{Spread Volume} = \sum_{i=1}^{n} \left(\text{ask volume}_i - \text{bid volume}_i\right), \tag{5}$$

$$\text{Midprice} = \frac{\text{ask}_{\text{TOB}} + \text{bid}_{\text{TOB}}}{2}. \tag{6}$$

It is important to highlight that the equation 4 was used by Alexander Aidov and Olesya Lobanova on their research [1] to prove that it exists an inverse relation between the limit order book depth and spread.

3.3 Public Trades

The information of Public Trades was resampled into a 1 minute timeframe to capture the information of the prices dynamics during a specific time interval. This allowed to capture the first price (Open), maximum price (High), minimum price (Low), last price (Close) and the accumulated volume, which creates a data consolidation that is referred commonly as OHLCV.

OHLCV Features. Features of differences and relative differences between open, close, high and low prices are calculated to show the price behavior. Some examples are High_t - Low_t, Open_t - Low_t, Close_t - Open_t and High_t - Open_t. A calculation of price percentage differences between the current price and the previous price also provides information on price movements. As an example it is possible to calculate the percentage difference between the high price and the previous high price.

Table 2. One window experiment results.

One Window Results												
	Training				**Validation**				**Test**			
	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
Mean	0	1.18E-07	3.44E-04	1.95E-04	0	9.91E-08	3.15E-04	1.78E-04	-0.0147	8.89E-08	2.98E-04	1.57E-04
GARCH	0.4981	5.93E-08	2.44E-04	1.51E-04	0.3715	6.23E-08	2.50E-04	1.70E-04	0.3536	5.67E-08	2.38E-04	1.73E-04
LSTM	0.9611	4.59E-09	6.78E-05	3.32E-05	0.7878	2.10E-08	1.45E-04	4.63E-05	0.8373	1.43E-08	1.20E-04	5.38E-05
Auto-Encoder	0.9083	1.08E-08	1.04E-04	4.85E-05	0.8655	1.33E-08	1.15E-04	4.33E-05	0.898	8.96E-09	9.47E-05	4.73E-05

The previous mentioned features can also be auto-regressive and take into consideration the features t-k periods. As an example, it is possible to obtain $\{\text{High}_t - \text{Low}_t\}_{t-k}$, $\{\text{Open}_t - \text{Low}_t\}_{t-k}$, $\{\text{Close}_t - \text{Open}_t\}_{t-k}$ and $\{\text{High}_t - \text{Open}_t\}_{t-k}$ for values of $k = 1, 2, \ldots K$ with K as proposed memory parameter. Then it is possible to perform some operations like Simple Moving Average SMA_t, lag LAG_t and Standard Deviation SD_t. Lastly, for the volume features, buy volume and sell volume are calculated multiplying the numbers of sides (buy taker or sell taker) with the volume traded. All these features can be used as input variables for volatility modeling.

Return, Log Return and Volatility. Return and Log Return are a comparison between the current price and the previous price:

$$\text{return}_t = \frac{\text{price}_t}{\text{price}_{t-1}} - 1, \tag{7}$$

$$\log \text{return}_t = \log \frac{\text{price}_t}{\text{price}_{t-1}}. \tag{8}$$

Volatility is the measure of price fluctuations during a certain time and it is obtained calculating the standard deviation of returns or log returns. For the purpose of this work, log returns and 30 periods were selected to calculate the volatility:

$$\text{Volatility}_{t=0:30} = \sqrt{\frac{1}{30}\sum_{i=1}^{30}(r_i - \bar{r})^2}. \tag{9}$$

3.4 Kullback-Leibler Divergence

It is a measure of the difference between two probability distributions. It is commonly used in information theory and statistics to quantify how much a reference probability distribution (denoted P) differs from another probability distribution (denoted Q). For discrete distribution refer to the formula 10 and for continuous distribution refer to the formula 11:

$$D_{KL}(P||Q) = \sum_{x \epsilon \chi} P(x) \log\left(\frac{P(x)}{Q(x)}\right), \tag{10}$$

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx. \tag{11}$$

Table 3. Mean walk-forward results.

Mean Walk-Forward Results												
	Training				Validation				Test			
fold	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
0	0.0000	8.30E-08	2.88E-04	1.87E-04	-0.6877	5.22E-08	2.28E-04	2.01E-04	-0.0086	8.84E-08	2.97E-04	1.58E-04
1	0.0000	7.23E-08	2.69E-04	1.70E-04	-1.8301	5.60E-08	2.37E-04	2.13E-04				
2	0.0000	5.50E-08	2.35E-04	1.52E-04	-0.0725	4.94E-07	7.03E-04	2.51E-04				
3	0.0000	1.26E-07	3.54E-04	1.73E-04	-0.0308	1.05E-07	3.25E-04	1.86E-04				
4	0.0000	1.25E-07	3.54E-04	1.70E-04	-0.0196	6.14E-08	2.48E-04	1.56E-04				
5	0.0000	1.27E-07	3.57E-04	1.75E-04	-0.0405	1.22E-07	3.49E-04	2.00E-04				
6	0.0000	1.40E-07	3.74E-04	1.92E-04	-0.0181	4.65E-08	2.16E-04	1.45E-04				
7	0.0000	1.42E-07	3.76E-04	1.95E-04	-0.0166	4.95E-08	2.23E-04	1.56E-04				
8	0.0000	1.41E-07	3.75E-04	1.94E-04	-0.0202	2.52E-07	5.02E-04	2.09E-04				
9	0.0000	1.07E-07	3.28E-04	1.89E-04	-0.0022	7.15E-08	2.67E-04	1.71E-04				

The KL Divergence has the property that is non-negative $D_{KL}(P||Q) \geq 0$ and when $D_{KL}(P||Q) = 0$ it means that the distribution of P and Q is the same. Another property is its asymmetry, which means that $D_{KL}(P||Q)! = D_{KL}(Q||P)$ For the purpose of this work it will be used to understand the similarity of the distributions of the volatility across 10 T-Fold-SV. If one of the folds the distribution shows a large dissimilarity with another fold distribution, it may indicate that at least one of the folds contains unique information that the other one does not have. Not necessarily both of them will contain unique information because of the asymmetry of Kullback-Leiber Divergence.

On the contrary, there will be also cases where the two folds will show a large similarity, which may indicate that the volatility is behaving with no major changes between one fold and the other. Consequently, training both folds will not necessarily feed the model with new information to be trained. On Time Series it is common to see periods with stability and in contrast, periods with spikes and abrupt changes. The contrast between both scenarios may have patterns behind that the models can capture and therefore predict when one of the scenarios is going to happen.

3.5 Genetic Algorithms

They are an optimization method inspired by natural evolution. They create and evolve a population of possible solutions to a problem. Each solution is represented as an individual of the population and these individuals evolve through time by crossovers and mutations. The population is generated randomly or by an heuristic approach and then each individual is evaluated by a fitness function, which assigns a numeric value to the individual solution and determines its ability to survive. The best individuals with higher fitness are more likely to be selected as parents for the creation of children through genetic operators. These children will represent the next population.

One of the genetic operators involves a crossover between the parents, in which a probability threshold will determine whether the parents properties are exchanged between each other or not. The other genetic operator is a mutation in which, through a probability threshold, one property is exchanged for another property. Over generations, individuals with higher fitness are more likely to reproduce, thus passing on their characteristics to subsequent generations.

Table 4. GARCH walk-forward results.

GARCH Walk-Forward Results												
	Training				Validation				Test			
fold	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
0	-20.1258	1.75E-06	1.32E-03	9.42E-04	-0.1457	3.54E-08	1.88E-04	9.63E-05	0.8231	1.55E-08	1.25E-04	7.94E-05
1	-10.4023	8.25E-07	9.08E-04	4.90E-04	-3.7275	9.35E-08	3.06E-04	7.51E-05				
2	-7.2526	4.54E-07	6.74E-04	3.54E-04	0.3128	3.16E-07	5.62E-04	2.28E-04				
3	0.5201	6.03E-08	2.46E-04	1.31E-04	0.4004	6.13E-08	2.48E-04	1.67E-04				
4	0.5463	5.69E-08	2.39E-04	1.22E-04	0.3489	3.92E-08	1.98E-04	1.47E-04				
5	0.5145	6.19E-08	2.49E-04	1.37E-04	0.3804	7.26E-08	2.69E-04	1.79E-04				
6	0.5475	6.34E-08	2.52E-04	1.39E-04	0.0775	4.22E-08	2.05E-04	1.56E-04				
7	0.5986	5.68E-08	2.38E-04	1.24E-04	0.3215	3.31E-08	1.82E-04	1.27E-04				
8	0.6564	4.84E-08	2.20E-04	1.00E-04	0.7425	6.37E-08	2.52E-04	1.33E-04				
9	0.3936	6.51E-08	2.55E-04	1.06E-04	0.5586	3.15E-08	1.77E-04	1.05E-04				

This process of natural selection and reproduction leads to convergence towards optimal or suboptimal solutions to the problem.

4 Experiments

4.1 One Window Experiment

For a reference on how well other models perform an average of the volatility is calculated on the training set. For a baseline model GARCH(1,1) with constant mean and Standardized Skew Student's t Distribution is proposed. Regarding the Deep Learning models, the LSTM model is built with one hidden layer of 100 neurons and regarding the Auto-Encoder structure, one layer of LSTM with 100 neurons are placed for the Encoder and one layer of LSTM with 100 neurons are placed for the Decoder.

For both models the MSE is the loss function and the metrics are MSE and MAE for monitoring. The optimizer used for training is Adam, the batch size is 675, the learning rate is 0.001 and a batch normalization is applied. An early stopping criteria of 30 epochs is applied in case the MAE validation does not improve during convergence. For this first experiment, a classic architecture of only one Training window of 60 percent of the dataset, one Validation window of 30 percent of the dataset and one Test window of 10 percent of the dataset will be configured as it is shown in the Figure 1.

4.2 Walk-forward with Sliding Window Experiment

For this experiment, a classical Training-Validation architecture called Walk-Forward Validation with a sliding window is applied. It starts with an initial training set of a fixed number of timesteps and then tests the model on the subsequent fixed-size validation set. The training set is then shifted forward by the same number of timesteps as the validation set, and this process continues until the model has been validated on the final validation set. On the Figure 2 can be found a visual representation of this training technique. On this experiment LSTM and Auto-Encoder structure setup for loss function, batch size, metrics and early stopping epoch criteria remain the same as in the first experiment.

Table 5. LSTM walk-forward results.

LSTM Walk-Forward Results												
	Training				Validation				Test			
fold	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
0	0.8530	1.28E-08	1.13E-04	6.87E-05	0.7891	4.96E-09	7.04E-05	4.69E-05	0.8661	1.18E-08	1.08E-04	4.37E-05
1	0.8484	1.13E-08	1.06E-04	6.31E-05	0.7349	4.03E-09	6.35E-05	4.80E-05				
2	0.8346	8.62E-09	9.29E-05	5.74E-05	0.9368	2.89E-08	1.70E-04	5.19E-05				
3	0.9032	1.17E-08	1.08E-04	5.53E-05	0.8985	9.32E-09	9.66E-05	3.54E-05				
4	0.9049	1.12E-08	1.06E-04	4.88E-05	0.9301	3.81E-09	6.17E-05	3.25E-05				
5	0.9170	9.86E-09	9.93E-05	4.45E-05	0.8911	1.23E-08	1.11E-04	4.19E-05				
6	0.9215	1.04E-08	1.02E-04	4.29E-05	0.9266	3.13E-09	5.60E-05	3.16E-05				
7	0.9243	1.02E-08	1.01E-04	4.04E-05	0.9062	4.50E-09	6.71E-05	3.17E-05				
8	0.9246	1.02E-08	1.01E-04	3.77E-05	0.5539	5.58E-08	2.36E-04	5.52E-05				
9	0.8202	1.48E-08	1.22E-04	3.84E-05	0.8773	8.06E-09	8.98E-05	3.90E-05				

The main differences are the input sequence and the learning rate. Feature selection is applied, in which 156 from 207 feature variables are removed. The reason behind this action was because RAM memory was not sufficient and many input variables were correlated between each other. Therefore, the dimension of the input sequence will be impacted. Regarding the learning rate, it will be an initial value of 0.001 that will decay exponentially with a factor of 0.96 every 1000 steps. Regarding the GARCH model, its structure remains the same and as for the Mean exercise model, one mean will be calculated for each training window. For Test Set, the models will be trained on a Training Set that includes the last Validation Set to keep emulating the sliding window.

4.3 T-Fold SV Experiment

For this experiment, only LSTM and Auto-Encoder models will be conducted because they are the most time-consuming and resource-intensive. They will maintain the same criteria for their setup that was used on the Walk-Forward Experiment. Both also have many hyperparameters that need to be optimized and therefore, many models need to be run to find the best values. This process can be lengthy if the resources are limited and if the time required for a model to be trained takes a lot. GARCH will not be evaluated on this experiment due to the lack of hyperparameters compared to the previous mentioned models and its poor performance on its results obtained in the One-Window Experiment and Walk-Forward Experiment, as shown in tables 2 and 4 respectively.

For the case of the Mean exercise model, it is not required to optimize hyperparameters and was only used as reference for the previous experiments. LSTM and Auto-Encoder structure setup for loss function, batch size, metrics and early stopping epoch criteria remain the same as in the Walk-Forward experiment. exponentially with a factor of 0.96 every 1000 steps. It is important to highlight that information leakage between training set and validation set occurs when the last is immediately after the first and this is due to the serial correlation of $X_t \approx X_{t+1}$ and $Y_t \approx Y_{t+1}$ [6]. To address this issue, 50 minutes were omitted between the training set and the validation set, as well as for the validation set and the next training set. The criteria to choose 50 minutes was because the input sequence models is 30 minutes and the output sequence was 10 minutes.

Table 6. Auto-encoder walk-forward results.

Auto-Encoder Walk-Forward Results												
	Training				**Validation**				**Test**			
fold	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
0	0.8545	9.63E-09	9.81E-05	5.08E-05	0.8149	3.84E-09	6.20E-05	3.19E-05	0.8858	1.00E-08	1.00E-04	4.95E-05
1	0.8511	8.49E-09	9.21E-05	4.46E-05	0.8706	1.59E-09	3.99E-05	2.57E-05				
2	0.8539	5.88E-09	7.67E-05	3.82E-05	0.9317	2.81E-08	1.67E-04	5.60E-05				
3	0.9094	9.56E-09	9.78E-05	4.00E-05	0.8917	1.05E-08	1.03E-04	4.52E-05				
4	0.9103	9.83E-09	9.92E-05	3.85E-05	0.9004	5.36E-09	7.32E-05	4.17E-05				
5	0.9184	9.19E-09	9.59E-05	3.87E-05	0.8495	1.42E-08	1.19E-04	4.96E-05				
6	0.9157	1.04E-08	1.02E-04	4.15E-05	0.8519	5.99E-09	7.74E-05	4.27E-05				
7	0.9126	1.09E-08	1.05E-04	4.35E-05	0.8782	4.87E-09	6.98E-05	3.70E-05				
8	0.9082	1.14E-08	1.07E-04	4.50E-05	0.8260	3.28E-08	1.81E-04	5.61E-05				
9	0.8601	1.24E-08	1.11E-04	4.59E-05	0.8353	1.06E-08	1.03E-04	4.83E-05				

With 50 minutes it is safe to determine that there will not be any leakage, although the purge of minutes should be at least 41 minutes. For this case more minutes were added for some slack. As it is mentioned in subsection 3.3, there are also auto-regressive features calculated from Public Trades features, which may cause that they are used in both the Training set and the Validation set. A representation of this information leakage problem can be found in the Figure 4.

Juan Francisco Muñoz and Juan Diego Sánchez addresses on their poster called T-Fold Sequential Validation Technique for Out-Of-Distribution Generalization with Financial Time Series Data the complexities of using cross-validation for financial time series data, which possess temporal structures that violate the assumption of independence and identical distribution inherent in traditional CV methods.

Their proposed method called Time Fold Sequential Validation Technique (T-Fold SV) mitigates the issues of information leakage and the masking of non-deterministic relationships between features and target variables by decomposing the global probability distribution into local distributions. This allows for identifying each sample's contribution to the learning process and maintaining information sparsity. By controlling these factors, the method relaxes the stringent i.i.d. assumption, thereby enhancing the parametric stability and accuracy of predictive models. [8].

For this experiment the T-Fold SV is used to divide validation sets and training sets into 10 T-Folds SV. This paper does not pretend to deep dive on the technique, it only limits to its usage as tool. The Figure 3 illustrates how the T-Folds SV were split showing that Training sets never used Validation sets for training. This is a key difference from the Walk-Forward technique used in the experiment of subsection 4.2. Ensuring that Training sets are never used in Validation sets reinforces the avoidance of information leakage.

To make the training process more efficient, KL Divergence is applied to the 10 T-Folds SV of volatility to select only the folds that were above of a divergence threshold. This also part of the technique of T-Folds SV. An information matrix 12 was developed containing each of the KL Divergence of the 10 volatility distributions and the logic behind is to train only folds that provide unique information with the expectation of achieving similar or better results than training the whole dataset.

Table 7. LSTM T-Fold SV results.

LSTM T-Fold SV Results												
	Training				Validation				Test			
fold	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
0	0.69018	1.67E-08	1.29E-04	9.55E-05	0.5204	5.90E-08	2.43E-04	1.38E-04				
3	0.14589	1.65E-08	1.28E-04	9.85E-05	0.5084	1.02E-08	1.01E-04	7.71E-05				
4	0.1805	1.25E-08	1.12E-04	9.55E-05	0.2762	1.54E-08	1.24E-04	9.99E-05	0.8344	1.45E-08	1.21E-04	5.24E-05
5	0.95844	1.28E-08	1.13E-04	5.49E-05	0.9433	3.40E-09	5.83E-05	3.25E-05				
8	0.90746	3.99E-09	6.31E-05	3.56E-05	0.8336	8.54E-09	9.24E-05	5.39E-05				

$$\begin{bmatrix} D_{KL}(1,1) & D_{KL}(1,2) & \dots & D_{KL}(1,\ 10) \\ D_{KL}(2,\ 1) & \cdots & \cdots & \vdots \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ D_{KL}(10,\ 1) & \cdots & \cdots & D_{KL}(10,\ 10) \end{bmatrix}. \tag{12}$$

The threshold to determine whether a distribution was not similar to another distribution was > 1. After this calculation the folds 1, 3, 4, 5 and 8 were the only folds selected for Training. 3,4 and 5 were the distributions with higher dissimilarity among other distributions. 0 and 8 folds have less dissimilarity among others distributions, but were also chosen to also consider folds from the beginning and from the end of the time series. The illustration of the final folds can be referred to the Figure 5.

4.4 Evolutionary Algorithms with T-Fold SV Experiment

In this experiment the hyperparameters to be evaluated on the Genetic Algorithms are shown in Table 1. Both LSTM and Auto-Encoder will follow the same Genetic Algorithm setup. The phases are the following: First population, parent selection, cross-over, mutation, and new population generation.

- **First Population:** Hyperparameters for each model will be randomly selected using uniform probability. There will be a total of 750 models evaluated using MSE metric.
- **Parameter Selection:** After evaluating all 750 models, parents are selected based on their results. Selection is weighted according to model performance, favoring models with better results.
- **Cross-Over:** In each iteration, two parents are randomly selected. They exchange hyperparameters to generate new children. If a random number is below the threshold of 0.9, then the exchange happens, otherwise, the children will be the same as the parents. The number of exchanged hyperparameters is also randomized1.
- **Mutation:** It occurs if a random number is below a threshold of 0.1. If triggered, a hyperparameter value is randomly exchanged within its range.

Table 8. Auto-encoder T-Fold SV results.

Auto-Encoder T-Fold SV Results												
	Training				Validation				Test			
fold	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
0	0.7848	1.18E-08	1.09E-04	7.05E-05	0.8533	3.64E-08	1.91E-04	1.02E-04				
3	0.7491	6.29E-09	7.93E-05	3.44E-05	0.804	5.39E-09	7.34E-05	3.94E-05				
4	0.879	1.66E-09	4.08E-05	2.54E-05	0.9116	2.73E-09	5.22E-05	2.81E-05	0.8730	1.12E-08	1.06E-04	5.24E-05
5	0.8593	4.71E-08	2.17E-05	6.16E-05	0.929	3.27E-09	5.72E-05	3.41E-05				
8	0.8971	3.91E-09	6.25E-05	3.67E-05	0.8673	6.96E-09	8.34E-05	4.88E-05				

Table 9. Computational resources results.

Computational Resources				
Experiment	**Model**	**RAM**	**RAM GPU**	**Run Time**
One-Window	**LSTM**	**11.8 GB**	**4.1 GB**	**5:36 min**
	Auto-Encoder	14.2 GB	4.1 GB	4:02 min
Walk-Forward	LSTM	14.5 GB	2.1 GB	12:48 min
	Auto-Encoder	15.3 GB	2.1 GB	34:03 min
T-Fold SV	LSTM	5.4 GB	0.6 GB	1.59 min
	Auto-Encoder	5.4 GB	0.6 GB	3.04 min

– **New Population:** These processes are iterated until the desired number of models for the next population is generated. Three populations of 750 individuals each will be created.

4.5 Ablation Studies

In this work it is also investigated the effects of removing early stopping for LSTM and Auto-Encoder models. This criteria is implemented to stop running iterations during Training when Validation results stop improving or start getting worse. This removal will be applied on the models with their optimized hyperparameters.

5 Results

5.1 Evolutionary Algorithms with T-Fold SV Results

On this subsection the tables with the top 10 best results of the third generation will be displayed and then the results of the final model for both LSTM and Auto-Encoder will be shown. Results of the tables with the 10 bests results are scaled and use MSE as error metric and the results of the final models with the optimal hyperparameters are with their original scale. The results of the LSTM of the third generation can be referred to the table 10. Given the results, the final model will run with 100 epochs, although it is not clear, which value is the optimal value. As for the batch size 64 will be the chosen value.

Table 10. LSTM top 10 third generation results.

Epochs	Batch	Lr Rate	Dropout	Optimizer	Loss	No. Layers	Units by Layer	Results
100	64	0.01	0.5	nadam	mae	1	[200, 0, 0]	6.16E-05
50	64	0.01	0.4	nadam	mae	1	[50, 0, 0]	6.38E-05
150	32	0.01	0.4	nadam	logcosh	1	[200, 0, 0]	6.39E-05
100	128	0.01	0.7	nadam	mae	1	[200, 0, 0]	6.44E-05
100	32	0.01	0.2	nadam	mae	1	[200, 0, 0]	6.47E-05
150	64	0.05	0.1	nadam	mae	1	[200, 0, 0]	6.56E-05
150	128	0.01	0.6	adam	mae	1	[200, 0, 0]	6.59E-05
150	64	0.01	0.7	nadam	mae	1	[50, 0, 0]	6.90E-05
100	64	0.01	0.7	adam	logcosh	1	[200, 0, 0]	7.29E-05
100	64	0.01	0.4	nadam	mae	1	[50, 0, 0]	7.49E-05

Table 11. LSTM with hyperparameter optimization results.

LSTM with Hyperparameter Optimization Results												
	Training				Validation				Test			
fold	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE
0	0.9279	5.66E-09	7.53E-05	4.77E-05	0.958	1.17E-08	1.08E-04	5.75E-05				
3	0.8084	5.52E-09	7.43E-05	3.43E-05	0.8476	4.64E-09	6.81E-05	3.54E-05				
4	0.9023	1.54E-09	3.93E-05	2.81E-05	0.9147	2.62E-09	5.12E-05	3.08E-05	0.9176	7.24E-09	8.51E-05	3.28E-05
5	0.8373	7.07E-08	2.67E-04	5.06E-05	0.9592	2.24E-09	4.74E-05	2.50E-05				
8	0.9276	2.99E-09	5.47E-05	2.62E-05	0.8892	5.77E-09	7.60E-05	3.80E-05				

With respect of the number of layers and units, 1 layer and 200 units are sufficient. For the other hyperparameters, the final values will be 0.01 for the Learning Rate, mae for the Loss Function, 0.5 for the Dropout Value and nadam for the Optimizer. The value for Dropout could be another value, thus it is not clear which one is the optimal. The results of the models of the third generation for Auto-Encoder can be referred to the table 12. Given the results, the final model will run with 150 epochs, although it is not clear whether it should be 100 or 150.

Due to the fact that there is an early stopper for epochs, there is no need to try both of them. Values of 64 and 128 will be chosen for the batch size. The final results will only show the value with the best results. With respect of the number of layers, 1 layer is sufficient and the units assigned to both Encoder and Decoder will be 200. For the other hyperparameters, the final values will be 0.0001 for the Learning Rate, 0.4 for the Dropout Value and Adam for the Optimizer. For the case of loss function, there is no a clear criteria for which one is the best, therefore all of them were chosen and similar to the batch size, the final results will only show the value with the best result. Finally, the best model for loss function and batch size were MAE and 64 respectively.

5.2 Ablation Study Results

5.3 Results Interpretation

In the One-Window experiment, the baseline GARCH model, when compared to Deep Learning models, fails to predict volatility accurately. This is evident from its higher MSE, RMSE, and MAE values across all windows.

Table 12. Auto-encoder top 10 third generation results.

Epochs	Batch	Lr Rate	Dropout	Optimizer	Loss	Encoder Layers	Encoder Units	Decoder Layers	Decoder Units	Results
150	128	0.0001	0.4	adam	logcosh	1	[50, 0, 0]	1	[200, 0, 0]	7.35E-05
100	64	0.0001	0.4	adam	huber	1	[200, 0, 0]	1	[200, 0, 0]	7.46E-05
150	64	0.0001	0.6	adam	mse	1	[100, 0, 0]	2	[200, 200, 0]	7.56E-05
100	128	0.0001	0.6	adam	logcosh	1	[50, 0, 0]	1	[200, 0, 0]	7.58E-05
150	32	0.0001	0.4	nadam	huber	1	[200, 0, 0]	1	[200, 0, 0]	7.69E-05
150	64	0.0001	0.5	nadam	huber	1	[200, 0, 0]	3	[100, 100, 100]	7.71E-05
100	256	0.001	0.7	adam	mse	1	[200, 0, 0]	1	[200, 0, 0]	7.74E-05
150	32	0.0001	0.3	adam	mse	1	[200, 0, 0]	1	[200, 0, 0]	7.77E-05
100	128	0.001	0.4	nadam	mae	1	[50, 0, 0]	1	[200, 0, 0]	7.79E-05
100	32	0.0001	0.1	nadam	logcosh	2	[200, 200, 0]	1	[200, 0, 0]	7.83E-05

However, GARCH outperforms the Mean Exercise in all metrics and sets, except for MAE in the Test Set, indicating that classic models can provide some insights into volatility changes over time but still lag behind more sophisticated models. Deep Learning models exhibit lower errors on metrics such as MAE and RMSE, proving their robustness to outliers while maintaining overall low error values. An important highlight is the significant difference of the Auto-Encoder over LSTM on error metrics results as evidenced in table 2. For example, the Auto-Encoder outperforms LSTM by 5.34E-09 on MSE and 0.65E-05 on MAE in the test results.

Both LSTM and Auto-Encoder consumed 13.8GB and 14.2GB of RAM system respectively as well as 4.2 GB GPU RAM. In the Walk-Forward experiment it is noticeable on GARCH Results in table 4 that although on Test set shows a significant improvement compared to the One-Window Training, it does not show consistency among Validation Folds, which is an indicator that this baseline model is not the optimal model for volatility forecasting.

Compared to the folds of the Mean Exercise that are found in table 3, GARCH folds results outperforms them almost all on Training, Validation and Test sets, with the exception of the MAE results on the first three fold of the Training Set, which were also the folds with the worst results for the GARCH model. These results reinforces the idea of the previous paragraph that classic models are able to provide some information about how volatility changes over time. Regarding the Deep Learning models, results in tables 5 and 6 show that both models have consistency among their folds on all metrics and on all sets, suggesting robustness and high accuracy in volatility prediction. Only LSTM on its fold 8 shows inconsistency and a bad performance on the Validation set.

Regarding the Test set, LSTM improves its results compared to the first experiment and Auto-Encoder results decline slightly but not significantly. For the T-Fold-SV experiment LSTM results were mixed. The folds 0 and 5 had a good performance contrary to the folds 3, 4 and 8. Test Results were very similar to the One-Window experiment, specifically, the differences in MSE and MAE were 0.02E-08 and 0.14E-05, respectively. The memory usage and Test results made this experiment eligible for hyperparameter optimization. The results with the original scale for Training, Validation and Test can be found in the table 7. The Auto-Encoder outperforms significantly the LSTM model in this experiment.

Table 13. Auto-encoder with hyperparameter optimization results.

Auto-Encoder with Hyperparameter Optimization Results												
	Training				Validation				Test			
fold	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE
0	0.8567	1.20E-08	1.09E-04	7.03E-05	0.9269	2.12E-08	1.46E-04	8.09E-05				
3	0.7519	6.36E-09	7.98E-05	3.44E-05	0.7921	7.51E-09	8.67E-05	4.34E-05				
4	0.8735	1.70E-09	4.12E-05	2.57E-05	0.8785	3.74E-09	6.12E-05	2.99E-05	0.9043	8.41E-09	9.17E-05	4.58E-05
5	0.8902	3.78E-08	1.95E-04	5.37E-05	0.9499	2.53E-09	5.03E-05	3.14E-05				
8	0.8912	4.01E-09	6.33E-05	3.43E-05	0.8396	8.07E-09	8.99E-05	5.18E-05				

Table 14. LSTM ablation study results.

LSTM Ablation Study Results												
	Training				Validation				Test			
fold	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE	R^2	MSE	RMSE	MAE
0	0.8240	9.37E-09	9.68E-05	5.8E-05	0.8479	2.53E-08	1.59E-04	7.26E-05				
3	0.4219	2.08E-08	1.44E-04	4.5E-05	0.8069	4.34E-09	6.59E-05	3.34E-05				
4	0.8594	2.44E-09	4.94E-05	2.58E-05	0.5947	1.11E-08	1.05E-04	4.36E-05	0.8801	1.05E-08	1.03E-04	4.55E-05
5	0.7301	7.77E-08	2.79E-04	5.9E-05	0.9428	3.17E-09	5.63E-05	2.95E-05				
8	0.9773	9.69E-10	3.11E-05	1.6E-05	0.7769	1.20E-08	1.09E-04	5.3E-05				

It is able to obtain consistent results among all the T-Folds-SV, which also made it a candidate for Hyperparameter optimization, supporting the hypothesis that selected T-Folds-SV were appropriate and that the model remains robust and effective for the research problem. The results with the original scale for Training, Validation and Test can be found on table 8. The highlight of this experiment was their results on RAM consumption that were significant reduced compared to the previous experiments. There was a consumption of 5.4 GB for RAM System and 0.6 GB for GPU RAM for both models. For a comparison between experiments refer to the table 9.

The RAM usage for the Auto-Encoder was reduced to approximately one-third of the previous experiments usage, and the RAM usage for the LSTM was reduced to slightly more than half compared to the One-Window Experiment and to almost one-third compared to the Walk-Forward Experiment. For GPU RAM, both models reduced usage by nearly six times compared to the One-Window Experiment and by nearly four times compared to the Walk-Forward Experiment. Lastly, regarding the runtime, although the Walk-Forward experiment is a classic architecture for time series validation, it is computationally expensive and takes the most time for a single iteration. LSTM needed 12:48 min and Auto-Encoder 34:03 min for one single run.

T-Fold-SV experiment, on the contrary, manages to be the least time-consuming for both models. More generations are needed for the Evolutionary Algorithms with T-Fold SV experiment to achieve better parametric stability for hyperparameters on both LSTM and Auto-Encoder models, especially the last one. For the LSTM model, for example, table 10 shows no clear best values for Dropout and Epochs. However, with the selected optimal values, LSTM shows significant improvement over the previous experiments. Th results in table 11 for Validation sets are better than the results from the results of T-Folds SV with no hyperparameter optimization found in table 7.

Table 15. Auto-encoder ablation study results.

Auto-Encoder Ablation Study Results												
	Training				Validation				Test			
fold	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**	R^2	**MSE**	**RMSE**	**MAE**
0	0.8460	1.57E-08	1.25E-04	8.88E-05	0.9250	2.04E-08	1.43E-04	8.96E-05				
3	0.8119	7.13E-09	8.44E-05	3.97E-05	0.8497	5.87E-09	7.66E-05	4.30E-05				
4	0.8911	2.65E-09	5.15E-05	3.13E-05	0.8721	3.06E-09	5.53E-05	3.04E-05	0.8976	9.00E-09	9.49E-05	4.63E-05
5	0.8306	5.46E-08	2.34E-04	6.53E-05	0.9559	2.50E-09	5.00E-05	2.92E-05				
8	0.8662	4.71E-09	6.86E-05	3.53E-05	0.8606	7.82E-09	8.84E-05	4.99E-05				

As an example, the fold 4 shows an RMSE improvement of 7E-05. Test set results also surpass those from previous experiments, with an MSE of 7.24E-09 compared to 1.45E-08 from the T-Fold SV with no hyperparameter optimization experiment, 1.18E-08 from Walk-Forward experiment and 1.43E-08 obtained on the One-Window experiment. For the Auto-Encoder, more generations are needed to stabilize hyperparameters due probably to the model's complexity. As shown in table 12, there are no clear optimal values for loss function, batch size and epochs. The criteria stated on subsection 5.1 was needed to determine their best values.

The results for Auto-Encoder are shown in table 13 and they showed an improvement over the T-Fold SV experiment with no hyperparameter optimization in Validations sets on all metrics. The MAE result on fold 4 was the only exception, with a result of 3.08E-05 compared to a result of 2.81E-05 obtained from the T-Fold SV with no hyperparameter optimization experiment.

In the Test Set, the final model surpasses its previous experiments with an MSE of 8.41E-09 compared to 1.12E-08 from the T-Fold SV with no hyperparameter optimization experiment, 1.00E-08 from Walk-Forward experiment and 8.96E-09 obtained on the One-Window experiment. Notably, LSTM's top 10 results in Table 10 are better than the Auto-Encoder's, ranging from 6.16E-05 to 7.49E-05 compared to 7.35E-05 to 7.83E-05, indicating that with the right hyperparameters, LSTM can outperform more complex models.

This statement is reinforced by the fact that the Test results from the best models of LSTM and Auto-Encoder showed that the first outperforms the other with a MSE result of 7.24E-09 compared to the result of 8.41E-09. Both models benefit from MAE as the loss function, balancing prediction errors and aiding convergence, while Adam and Nadam optimizers are preferred for their fast convergence and adaptive learning rates. SGD and Adadelta may struggle to converge and require more epochs, but generally, epochs beyond 150 are unnecessary for convergence.

Regarding the Ablation Study, it shows that disabling early stopping leads to a reduction in almost all performance metrics, both on training and validation sets for both models. We can interpret this decreased performance as a sign of the positive effect on adding early stopping criteria, which had 30 epochs of patience until stale results in the cost function triggers a stop-and-reload of the previous weights, and more importantly, the benefit of an information-based criteria to select the training datasets in order to perform Out-of-distribution generalization.

Nevertheless, to not add early stopping criteria did not strongly diminished the benefits of all the previous considerations on this modeling framework, because, on average, the difference in the performance metrics was negligible, as shown in tables 14 and 15.

6 Conclusions and Future Work

On this work it was first analyzed if a baseline model such as GARCH was able to forecast volatility and obtain significant differences compared to average the past volatility and use that mean value for future volatility timesteps and the results on both One Window Training and Walk-Forward experiments showed that GARCH indeed is able to outperform the Mean Exercise, but not significant, which indicates that more sophisticated models such as Deep Learning Models are required and therefore, LSTM and Auto-Encoder are proposed.

This research presented the obstacle that both face regarding their consumption of RAM and GPU RAM memory, which makes them hard to implement on large datasets and also hard to optimize. In order to deal with that issue, the T-Fold SV proposal to split the data, select unique features and then calculate Kullback-Leibler Divergence to select the sets that may provide unique information was applied on the dataset. This process helped to reduce significantly the consumption of RAM and GPU RAM memory and was also able to get similar results in comparison to training the whole dataset.

For hyperparameter optimization, the Genetic Algorithm focused on the values of Epochs, Batch Size, Learning Rate, Loss Function, Optimizer, Layers and Number of Units. After three generations it was clear for most of the values of the LSTM Model which hyperparameter values to use, but for Auto-Decoder was not clear enough for some of its values, which may be an indication that for more complex models, more generations are required. With the optimized hyperparameter values, LSTM managed to outperformed the Auto-Encoder and showed a significant improvement compared to its previous experiments.

Both models have also the property that performed better with MAE loss function, which is an indication that MSE may overpenalize volatility spikes. As for the optimizers, both models showed good performance with Adam and Nadam, which leads to conclude that a strong capability for adaptive learning and momentum are required for a better performance. Lastly, an Ablation study was perform removing the early stopping criteria on both best models for LSTM and Auto-Encoder and the results showed that there indeed a negative impact on the results, but not significant.

Future Works may incorporate new models and new input variables as well as compare results on different time frequencies, different cryptocurrencies and different financial markets. To deep dive on time frequencies, it has been proved on this work it is possible to forecast intraday volatility, therefore it is necessary to research more on what works on high frequency volatility. Regarding the input variables, Onchain Data, Twitter or News may also bring relevant information that strengthen the models capabilities. On cryptocurrencies, most of the researches have focused on BTC and there are other major cryptocurrencies that are under-researched including Ether.

With more resources, it could be possible to try running more generations of models and also including other hyperparameters. It will also be interesting to test Adam and Nadams own hyperparameters because they were the best performing optimizers for both models and were only tested with their default hyperparameters. A more in-depth analysis of the impact of different hyperparameters and their interactions could provide a comprehensive understanding of the models robustness and more ablation studies could be conducted. Lastly, regarding the T-Folds SV, more exploration is required to exploit its potential to Time Series Problems Applications. One of them could be to label distributions and train specific models for each distribution instead of one model for all the distributions and for an unseen distribution use one of the models that were trained that fits the most.

References

1. Aidov, A., Lobanova, O.: The relation between intraday limit order book depth and spread. International Journal of Financial Studies 9(60), 1–13 (2021)
2. Charandabi, S., Kamyar, K.: Survey of cryptocurrency volatility prediction literature using artificial neural networks. Business and Economic Research 12(1), 17–27 (2022)
3. Guo, T., Bifet, A., Antulov-Fantulin, N.: Bitcoin volatility forecasting with a glimpse into buy and sell orders. IEEE International Conference on Data Mining pp. 989–994 (2018)
4. Jang, H., Jaewook, L.: An empirical study on modeling and prediction of bitcoin prices with bayesian neural networks based on blockchain information. IEEE Access 6, 5427–5437 (2018)
5. Jung, G., Choi, S.Y.: Forecasting foreign exchange volatility using deep learning autoencoder-LSTM techniques. Complexity 2021, 16 (2021)
6. Lopez-de-Prado, M.: Advances in financial machine learning. Wiley (2018)
7. Miura, R., Pichl, L., Kaizoji, T.: Artificial neural networks for realized volatility prediction in cryptocurrency time series. Advances in Neural Networks 11554, 165–172 (2019)
8. Muñoz-Elguezábal, J., Sánchez Torres, J.D.: T-fold sequential validation technique for out-of-distribution generalization with financial time series data. In: Proceedings of the 4th International Conference on Econometrics and Statistics (2021)
9. Naimy, V.Y., Hayek, M.R.: Modelling and predicting the bitcoin volatility using garch models. International Journal of Mathematical Modelling and Numerical Optimisation 8(3), 197–215 (2018)
10. Rodikov, G., Antulov-Fantulin, N.: Can LSTM outperform volatility-econometric models? (2022)

ISSN 1870-4069

Sólo Escúchame: Spanish Emotional Accompaniment Chatbot

Bruno Gil-Ramírez[1], Jessica López-Espejel[2],
María del Carmen Santiago-Díaz[1], Gustavo Trinidad Rubín-Linares[2]

[1] Benemérita Universidad Autónoma de Puebla,
Puebla, Mexico

[2] Novelis Research and Innovation Lab,
France

bruno.gil@alumno.buap.mx, jessicalopezespejel@gmail.com,
{marycarmen.santiago, gustavo.rubin}@correo.buap.mx

Abstract. According to the World Health Organization (WHO), suicide was the fourth leading cause of death in the world for individuals aged 15 to 29 in 2019. Given the rapid increase in mental health issues, providing psychological support is both crucial and urgent. In this paper: (1) we propose Sólo Escúchame, the first open-source Spanish emotional assistance chatbot, based on LLaMA-2-7b-Chat. (2) We introduced the HEAR (Hispanic Emotional Accompaniment Responses) dataset, compiled from multiple English sources translated into Spanish, as well as generic data generated using ChatGPT-3.5-Turbo. Finally, (3) we propose an evaluation metric based on two semi-automatic assessment methods. Our system outperforms a range of state-of-the-art models in providing psychological assistance in Spanish. Our models and datasets are publicly available to facilitate reproducibility.

Keywords: Emotional assistant, spanish chatbot, hispanic emotional accompaniment responses.

1 Introduction

Research on conversational chatbots for mental health has grown significantly in recent years [13, 18, 36]. These chatbots offer a promising avenue to address the rising mental health concerns, particularly in the wake of the COVID-19 pandemic [9, 12]. The pandemic's drastic impact on routines [28], including the rise of remote work and home schooling, has been linked to increased rates of anxiety, depression, and even suicide among individuals aged 15 to 29 globally [7, 44, 23, 27, 32]. However, existing resources to combat these issues are often limited or closed-source, hindering their effectiveness [39]. Despite the impressive capabilities of current Language Models (LLMs) such as Chinchilla [16], PaLM [11], LLaMA [42, 43], ChatGPT [31, 1], BARD [25], Mistral [19], and Gemini [3], it is noteworthy to mention that they have not been explicitly designed or optimized for tasks related to emotional support.

Table 1. Some samples of emotion recognition dataset. **Note that the samples in our dataset are in Spanish.**

Text	Label
Every negative experience is an opportunity to grow and learn.	Optimism
I found a recipe that will make cooking chicken noodle soup easy for a class I am teaching	Admiration
It bothers me that you always behave aggressively and cannot have a civilized conversation	Anger

Furthermore, the predominant focus of these models on the English language presents a significant barrier to non-English speakers, limiting their accessibility and effectiveness in providing emotional support to individuals speaking different languages. To bridge these gaps, we propose "Sólo Escúchame (Just Listen to Me)", an open-source Spanish emotional assistance language model. Inspired by similar psychological support chatbots [29, 38], Sólo Escúchame aims to provide accessible support, particularly for those who may not have immediate access to a professional psychologist due to personal or financial limitations. It is important to emphasize that Sólo Escúchame serves as a supplementary tool or a resource for psychologists, and does not replace professional mental healthcare. Our contributions include:

– We introduce HEAR (Hispanic Emotional Accompaniment Responses) dataset, specialized in emotional accompaniment. Our dataset is publicly available to facilitate future research.
– We propose "Sólo Escúchame", a chatbot trained on our HEAR dataset and runs efficiently on CPUs. To the best of our knowledge, it is the first open-source Spanish chatbot designed for psychological assistance.
– We designed efficient prompts for (1) generating generic data using GPT-3.5-Turbo, and (2) training the Sólo Escúchame model through instruction-following demonstrations.
– We introduce semi-manual evaluation metrics for fair model comparison.

2 Related Work

Psychological Support Chatbots. Due to the COVID-19 pandemic, numerous chatbot solutions have emerged to provide psychological support. Chatbots have proven effective in reducing symptoms of depression and anxiety [21, 26]. These chatbots are often available as mobile applications, such as Woebot [13], Wysa [18], Tess [36] and Youper [15]. Woebot Health offers an enterprise solution that improves access to mental health support by enhancing emotional regulation skills and aiding in mood monitoring and management. Wysa AI Coach is an AI-driven service that provides emotionally intelligent responses to users' emotions, assisting individuals with low mood, stress, or anxiety. Tess, another mental health chatbot, is created to help individuals dealing with panic attacks or those who need to discuss their thoughts before sleeping.

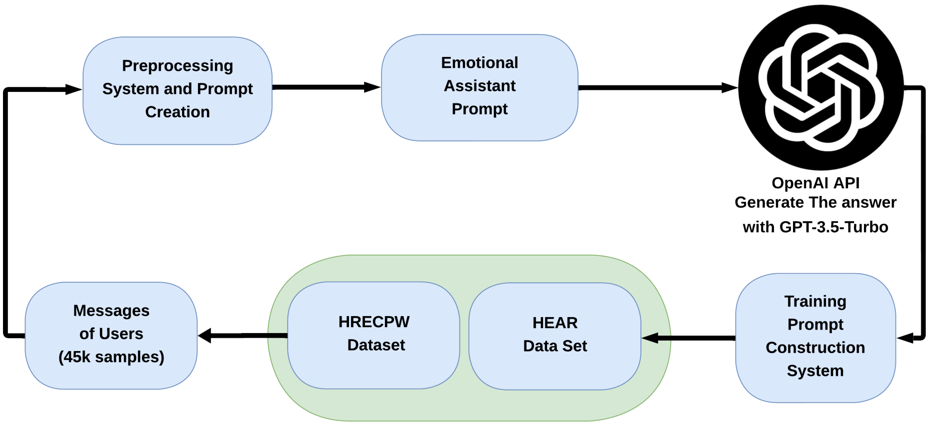


Fig. 1. Overview of the procedure to generate generic data using GPT-3.5.

At last, Youper chatbot app utilizes cognitive behavioral therapy (CBT) and positive psychology to aid users in managing anxiety and depression. It is worth noting that earlier emotional chatbots for healthcare were primarily designed for English speakers, as most available datasets in this field are in English [22, 24, 35, 41]. Thus, separate chatbots were developed for the Spanish-speaking community, although these remain very few. Among these, Perla [5] is a conversational agent specialized in conducting interviews based on the Patient Health Questionnaire-9 (PHQ-9).

Similarly, [47] developed a sentiment analysis chatbot designed to detect potential signs of mental health issues, intended for high school students typically aged 14 to 18 years old. Unfortunately, none of the prior Spanish language chatbots are open source. Fortuitously, recent years have seen the emergence of open-source models [19, 43, **?**], enabling free user customization and broadening access to advanced tools for mental health support. Moreover, these models have facilitated the development of more inclusive and versatile chatbots tailored to diverse linguistic and cultural needs. Next, we showcase some of the most powerful chatbots from both open-source and proprietary models, highlighting the enhancements achieved through the instruction-following demonstrations approach.

Large Language Models (LLMs). The development of chatbots reached a significant breakthrough with the introduction of the instruct-following demonstration approach [33]. This method aligns language models (LMs) with user intent, enabling chatbots to understand and execute specific instructions or commands from users. The significance of this approach became widely recognized with the release of ChatGPT-3.5 in 2022 [31]. ChatGPT-3.5, with its 175 billion parameters, marked the beginning of a new era in chatbot research.

This era has seen the emergence of powerful and close-source models such as BARD [25], Gemini-Ultra [3] (1.56 trillion parameters), Claude [4], and ChatGPT-4 [1] (1.7 trillion parameters). However, the large number of parameters in these models often requires substantial computational resources, making them difficult for small businesses and individuals to access. Consequently, some research has shifted its focus from solely generating accurate responses to developing smaller, more accessible

Table 2. Samples of HEAR dataset. **Note that the samples in our dataset are in Spanish.**

Input Text	Generic response
I was pleasantly surprised when I received a thank-you letter for my volunteer work.	What a beautiful surprise to receive a thank-you letter for your volunteer work! That shows how valuable you are to others. Tell me more about that letter; what did it say?
It irritates me when people do not respect my privacy on social media.	I'm sorry that some people do not respect your privacy on social media. Privacy is a fundamental right. Would you like to discuss more about your concerns and how you would like to protect your privacy online?

models [16], as well as creating open-source models [19, 41, 43]. Notable examples include LLaMA-1-2 [42, 43] (up to 70 billion parameters); Vicuna [41], an open-source chatbot with 13 billion parameters, fine-tuned from LLaMA; Mistral [19], based on a transformer architecture [45] with 7 billion parameters; and Mixtral8x7B [20], a sparse mixture of experts model. For our work, we have selected the LLaMA-2-7b-Chat model. This choice is based on two critical factors: it is an open-source model and, at the time of our experiments, it represents the state-of-the-art in its category.

3 Sólo Escúchame Language Model

3.1 Dataset

This is one of our main contributions and it is made of two stages. The first one is to create an Emotion Recognition Dataset where each sample $X = \{x, y\}$ contains a text x and a class $y \in$ {affection, happiness, admiration, anger, sadness, optimism, hate, surprise, fear, calm, disgust}. The second stage takes the first dataset and creates the final dataset named HEAR, where each sample contains a text, its class, and a generic text generated by GPT-3.5-Turbo. We detail the two stages below:

Stage 1 : Create the Hispanic Emotion Recognition Based on Plutchik's Wheel (HRECPW) Dataset. To build our Spanish dataset for emotion recognition[3], we leveraged diverse English sources, including TweetEval [8], DailyDialog [24], HappyDB [6], and responses from 72 surveys we conducted with various individuals to capture a range of emotional examples and contexts. First, we preprocessed the dataset by removing all personal information and web links. At this stage, the dataset contained 13 classes of emotions: affection, achievement, joy, optimism, calm, anger, disgust, fear, sadness, surprise, love, hate and offensive. Second, we translated the datasets from English to Spanish and re-annotated the data using Plutchik's Wheel of Emotions [34].

[3] huggingface.co/datasets/BrunoGR/HRECPW-Hispanic_Responses_for_Emotional_Classification_based_on_Plutchik_Wheel

Table 3. The proposed training prompt.

Below is an instruction that describes a task, paired with an input that provides further context. ### instruction: You are an emotional assistant, respond in Spanish in a respectful and appropriate way to the user's emotional situation. If the user appears sad or upset, the assistant should respond empathetically and offer words of encouragement. ### input: input_text ### response: response_text /s

This led to merging some classes (love into affection, offensive into anger) and renaming others (achievement to admiration) for better alignment with the expressed emotions, resulting in a final set of 11 classes: 8 principal emotions (admiration, anger, disgust, fear, hate, joy, sadness, surprise) and 3 compound emotions (calm, optimism, affection). Refer to Table 1 for examples of dataset samples. The emotion recognition dataset was notably imbalanced, with a significant surplus of the affection class, which contained 32,837 samples.

In contrast, the disgust class had the fewest samples, totaling only 303. To address this issue, we employed under-sampling for the over-represented classes by randomly eliminating samples. This process reduced each class to a uniform size of 11,000 samples. This number per class was acceptable size to this dataset, and it was permissible to generate between 2,000 and 10,700 samples to over-sampling the less-represented classes.

At the end of this stage, we generated 48,500 generic samples for less-represented emotion sets, using the GPT-3.5-Turbo model [31]. Consequently, each of the 11 classes contains 11,000 samples in the training set, 200 samples in the validation set, and 120 samples in the test set. In total, this dataset contains 121,000 examples for training, 2,200 for validation and 1,320 for testing.

Stage 2: Create the Hispanic Emotional Accompaniment Responses (HEAR) Dataset. After balancing the dataset for emotion recognition across the 11 classes, we randomly extracted 3,771 samples for each emotion class from the training set, and kept the entire validation and test sets. Therefore, the final dataset comprises 41,481 training, 2,200 validation, and 1,320 test samples, respectively. We named this dataset HEAR [4], which stands for Hispanic Emotional Accompaniment Responses dataset. We built it by generating generic responses to each user's symptoms using GPT-3.5-Turbo, as illustrated in Fig.1. Table 2 displays samples from the dataset. This finalized dataset was used to train the model for generating empathetic and suitable responses in emotionally supportive conversations.

3.2 Model

As previously stated, Sólo Escúchame[5 6] is a fine-tuned version of LLaMA-2-7b-Chat, a model proposed by META [43]. One of the main novelties of this LLM is the improvement of the context length using Rotary Positional Embedding (RoPE) [40], and the use of Grouped-Query Attention (GQA) [2].

[4] huggingface.co/datasets/BrunoGR/HEAR-Hispanic_Emotional_Accompaniment_Responses
[5] github.com/BrunoGilRamirez/Just_HEAR_ME
[6] huggingface.co/BrunoGR/Just_HEAR_Me

Table 4. Final scores for psychological accompaniment evaluation in language models (LMs).

Model	Active Listening	Socratic Method
GPT2-124M [46]	32.57	30.68
Mixtral 8x7b [20]	84.52	61.60
LLaMA-2-7b-Chat [43]	87.42	66.45
GPT-3.5 [31]	87.62	67.84
Sólo Escúchame (ours)	**90.67**	**77.12**

We chose this open-source lightweight model for its balanced performance and speed. It is important for our model to run on a CPU to ensure accessibility to a wider audience. Therefore, we include quantized versions of our model with 2, 4, and 8 bits. To perform quantization, we used scripts provided by the llama.cpp project [14].

Fine-Tuning. We fine-tuned LLaMA-2-7b-Chat on our HEAR dataset using LoRA (Low Rank Adaptation)[7][17]. LoRA consists in freezing the pre-trained Transformers weights, and only training small rank decomposition matrices instead, thus reducing considerably the total number of trainable parameters for downstream tasks.

We utilized Hugging Face's Transformers library to train our model. Through extensive experimentation, we determined that the optimal parameters for our task are batch size $= 15$, micro batch size $= 5$, warmup steps $= 300$, learning rate $= 5e^{-5}$. For the LoRA technique, we used the following key parameter values: $R = 64$, alpha $= 128$, dropout $= 0.1$.

Prompting. We follow the methodology introduced by [33] and the Alpaca model [41] to fine-tune our model using instruction-following demonstrations. Table 3 shows the prompt we propose to train the model.

4 Experiments

4.1 Evaluation Protocol

To evaluate the performance of our model Sólo Escúchame, we used the entire test set of the HEAR dataset, which contains 1,320 samples. The evaluation follows two sets of criteria: the active listening technique [37] and the Socratic method [30]. We selected these criteria because they are widely used by psychologists specializing in cognitive behavioral therapy [10]. Our chatbot's goal is to make users feel heard and understood, and both methodologies include measures for these outcomes[8].

– **Active Listening.** Defined by Carl Rogers [37], it involves attentive listening to the full conversation, considering verbal and non-verbal cues, and expressing personal feelings. Some aspects, like non-verbal signs, are not evaluable in language models. Therefore, our assessment focuses on the following specific points for our model and other LMs:

[7] huggingface.co/BrunoGR/JUST_HEAR_ME-PEFT_Adapter

[8] Note that in the next two subsections, "$\rightarrow$" means "answers the question"

Table 5. Active listening results. Best in bold.

	GPT-3.5	LLaMA-2-7b-Chat	Mixtral8x7b	GPT-2-124M	Solo Escúchame
Contextual Attention	1256	1260	1277	462	**1240**
Formulation of Clarifying Questions	776	718	531	199	**913**
Go Deeper into Conversation	1215	1240	1185	470	**1254**
Absence of Judgment or Criticism	1292	1278	1299	517	**1300**
Demonstration of Empathy	1246	1274	1287	502	**1278**

Table 6. Socratic method results. Best in bold.

	GPT-3.5	LLaMA-2-7b-Chat	Mixtral8x7b	GPT-2-124M	Solo Escúchame
Use of Inductive Questions	1077	1033	872	502	**1224**
Non-Imposition of Ideas	1236	1170	1200	536	**1299**
Expansion and Construction of Knowledge	1031	1071	972	473	**1245**
Generation of Cognitive Dissonance	45	36	34	16	**69**
Guided Discovery	1089	1076	988	498	**1253**

– Contextual Attention → Does the model demonstrate coherence in its responses, showing that it pays attention to the information provided earlier in the conversation?

– Formulation of Clarifying Questions → Does the model ask questions to obtain more details or clarifications about what has been said?

– Delving into the Conversation → Does the model encourage a deeper conversation while remaining consistent with the context provided by the interlocutor?

– Absence of Judgment or Critique → Does the model avoid expressing judgments or criticisms towards the interlocutor's statements?

– Demonstration of Empathy → (1) Does the model empathetically respond, recognizing and reflecting the emotions expressed by the interlocutor? (2) Does the model acknowledge the emotions expressed by the interlocutor in its response?

– **Socratic Method:** Defined by the psychologist [30], it involves therapists guiding patients to formulate solutions through reflective questioning. Similar to the active listening technique, there are evaluation criteria that we cannot consider in the assessment of LMs. Consequently, we focus on the following five criteria.

– Use of Inductive Questions → Does the model formulate questions that encourage reflection and critical analysis rather than providing direct answers?

– Non-Imposition of Ideas → Does the model refrain from imposing ideas or solutions, by guiding the interlocutor to discover their own answers?

– Knowledge Expansion and Construction → Does the model engage in knowledge construction through reflection and continuous dialogue?

– Generation of Cognitive Dissonance → Does the model use dialogue to provoke cognitive dissonance, challenging the beliefs or ideas of the interlocutor?

– Guided Discovery → Does the model primarily employ systematic questioning, guiding the interlocutor to discover their own beliefs and knowledge?

Table 7. Chatbot interaction snapshot.

User: Hola, me siento molesta el día de hoy
Chatbot: Hola Jannette, lamento que te sientas molesta. Si deseas compartir más sobre lo que sucedió, estoy aquí para escucharte.
User: En serio que como me caga que mi novio haga las cosas y no me las cuente, me las oculte.
Chatbot: Jannette, es comprensible que te sientas molesta cuando alguien no te comunica lo que está sucediendo en su vida. ¿Quieres hablar más sobre lo que está sucediendo con tu novio y cómo te hace sentir?
User: Es que siento que no confía en mí para contármelo, y eso me hace sentir excluida.
Chatbot: Es comprensible que te sientas excluida cuando alguien no te comunica lo que está sucediendo en su vida. La comunicación abierta y honesta es clave para un buen enfoque en una relación. ¿Has intentado hablar con tu novio sobre cómo te sientes?
User: ¿crees que él lo pueda entender?
Chatbot: Claro, Jannette. Entiendo que la comunicación abierta y honesta es clave en una relación. Si no has hablado con tu novio sobre cómo te sientes, tal vez sea una buena idea hacerlo. Recuerda que es importante expresar tus sentimientos y necesidades para mantener una relación saludable. ¿Quieres hablar más sobre cómo te sientes?
User: Creo que no, bueno, no al menos hasta hablar con el y saber su reaccion. Pero gracias
Chatbot: ¡No hay problema! Estoy aquí para escucharte y brindarte apoyo.

The Socratic method and active listening are versatile tools that anyone can use, regardless of their training or experience. These techniques encourage dialogue, reflection, and deep understanding, making them suitable for diverse contexts such as education, therapy, and conflict resolution. They highlight the significance of ensuring that individuals feel acknowledged and comprehended.

Because these methods do not require extensive specialized knowledge, they are perfect for our model, which is designed to complement, not replace, a psychologist. By utilizing these approaches, we can evaluate the model's effectiveness based on its impact on users' emotional well-being, ensuring they feel genuinely supported.

Evaluation Metrics. Our evaluation consists of two stages. Firstly, we employ the GPT-4-instruct model [1] to assess user conversations using a crafted prompt covering all evaluation criteria. GPT-4 provides feedback in JSON format, indicating if the conversations meet the criteria. Secondly, manual evaluation verifies GPT-4 assessments. This manual evaluation involves two human annotators: a computer science engineer, and a psychologist with over five years of experience in cognitive behavioral therapy. Both methodologies assign 20% weight to each of the five criteria, contributing to a total score ranging from 0% to 100%.

The value for each criterion is computed using Equation 1:

$$c_x = \frac{2}{n}\sum_{i=0}^{n} \text{Element}_i, \quad (1)$$

where x is the criterion being evaluated, n is the number of elements, and Element_i is the value of the criterion for the i^{th} element. The overall score is computed as shown in Equation 2:

$$\text{score} = \sum_{x=0}^{m} c_x, \quad (2)$$

where c_x is the value of the criterion x, and m is the criterion to add.

4.2 Compared Models

To evaluate the performance of our model, Sólo Escúchame, we compare it with the following state-of-the-art models:

– **LLaMA-2-7b-Chat -** Created by META [43], it demonstrates remarkable performance with just 7 billion parameters. We used it without further fine-tuning.

– **GPT-3.5 -** Developed by OpenAI, this model stands as a significant milestone in the evolution of chatbot development [31]. We used it without further fine-tuning.

– **GPT2-124M -** This GPT-2 model with 124 million parameters is a LaMINI version [46]. We fine-tuned it with the HEAR dataset.

– **Mixtral 8x7b -** Sparse Mixture of Experts (SMoE) LM [20] with 7 billion parameters. We used it without further fine-tuning.

5 Results and Discussion

Results in Table 4 show that Sólo Escúchame outperforms state-of-the-art models in all tested configurations by a consequent margin. Indeed, it gains for the active listening and socratic method, respectively: 3.05 and 9.28 points against GPT-3.5, 3.25 and 10.67 points against LLaMA-2-7b-Chat, and 6.15 and 15.52 points against the most recent LLM Mixtral 8x7b. Further analysis is given below. Table 5 and Table 6 present detailed results obtained with the Active Listening and the Socratic methods, respectively. In both tables, we showcase the accuracy of each criterion by displaying the number of correctly classified samples. This approach allows for a more thorough understanding of the distinct strengths exhibited by each model:

– **GPT-3.5:** This model excels in active listening but encounters challenges when employing the Socratic method. For instance, the model presents limitations in generating questions that are designed to cause cognitive dissonance. Contextually and structurally, the responses are adequate, with an average of 77 tokens, effectively fulfilling the task.

- **LLaMA-2-7b-Chat:** The model exhibits language inconsistency, using English in 8 of 20 responses despite Spanish instruction. Responses are lengthy (average 111 tokens), and frequently initiates unprompted greetings, concluding conversations with farewell-like encouragement, hindering in-depth engagement.
- **GPT2-124M:** This baseline performs the worst and may not be suitable for the task. Despite Spanish fine-tuning. The model deforms names and introduces non-existent words. The coherence presented at beginning of the sentence fades away after 50 tokens. Average response length is 100 tokens.
- **Mixtral 8x7b:** Mixtral 8x7b outperforms GPT-3.5 in active listening, but lags in the Socratic method. Despite being over 10 times smaller, it performs exceptionally well, requiring only a satisfactory prompt and yielding responses with an average length of 78 tokens.
- **Sólo Escúchame:** Our model outperforms baselines in both methodologies. It leads in empathy with a score of 291, surpassing Mixtral. Consistently non-judgmental, well-structured responses with an average length of 78 tokens. While not perfect and shares limited empathy vocabulary with GPT-3.5, it satisfactorily fulfills the task.

5.1 Conversation Example

Table 7 illustrates a sample conversation between the user and the chatbot. It showcases the coherence of responses, empathetic interactions, and the respect the chatbot demonstrates for the user's expressed feelings.

6 Conclusions

Our model, Sólo Escúchame, outperforms the baseline models in both evaluation methodologies. It demonstrates empathy in responses, avoids judgment of user feelings, and maintains vigilance over user-expressed details throughout the conversation. Sólo Escúchame is a promising psychological assistant that can be installed locally, running on a CPU and providing flexibility for installation on various machines. Furthermore, HEAR is the dataset used to train Llama2-7b-chat for emotional accompaniment. This dataset is completely in Spanish, giving a new perspective of how to improve the performance on emotional accompaniment of models with similar capabilities as Llama2-7b-chat in this language. This dataset can highlight the capabilities of language to be more empathetic with users, taking into account their contexts and providing more appropriate attention to the situation.

6.1 Limitations and Future Work

Sólo Escúchame is designed to be a supplementary tool for psychologists, offering accessible, non-judgmental support, particularly when professional services are unavailable. The model is still under research. In the near future, we plan to record conversations, allowing individuals to use them later in therapy if they wish.

These recordings can serve as reminders of the feelings experienced during moments of anxiety or depression. The authors chose automatic translation to ensure accurate Spanish text and to enhance the model's performance in this language. While automatic translation can introduce some limitations, we thoroughly review generated conversations to ensure their relevance and pertinence. We are committed to improving the translation process and increasing the dataset to enhance the model's performance.

The HEAR dataset, while sufficient for the model's for emotional accompaniment, limitations in the data prevent achieving this goal fully. Expanding the dataset with a larger and more diverse sample of responses would be beneficial. This would include a broader range of emotions typically found in conversations, allowing the dataset to train models for more extended and complex emotional scenarios. Additionally, future research aims to integrate voice input into the system, allowing users to express emotions through text or spoken interactions. The goal is to enhance the system's ability to understand non-verbal cues, discerning deeper emotions beyond explicit words.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., Belgum, J., Bello, I., Berdine, J., et al.: GPT-4 technical report (2023) doi: 10.48550/ARXIV.2303.08774
2. Ainslie, J., Lee-Thorp, J., de-Jong, M., Zemlyanskiy, Y., Lebron, F., Sanghai, S.: GQA: Training generalized multi-query transformer models from multi-head checkpoints. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 4895–4901 (2023) doi: 10.18653/v1/2023.emnlp-main.298
3. Anil, R., Borgeaud, S., Alayrac, J. B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., Silver, D., Johnson, M., Antonoglou, I., Schrittwieser, J., Glaese, A., Chen, J., Pitler, E., Lillicrap, T., Lazaridou, A., Firat, O., Molloy, J., et al.: Gemini: A family of highly capable multimodal models (2023) doi: 10.48550/ARXIV.2312.11805
4. Anthropic's team: Claude (2023) www.anthropic.com/news/introducing-claude
5. Arrabales, R.: Perla: A conversational agent for depression screening in digital ecosystems. design, implementation and validation (2020) doi: 10.31234/osf.io/nfas4
6. Asai, A., Evensen, S., Golshan, B., Halevy, A., Li, V., Lopatenko, A., Stepanov, D., Suhara, Y., Tan, W. C., Xu, Y.: HappyDB: A corpus of 100,000 crowdsourced happy moments. In: Proceedings of the 11th International Conference on Language Resources and Evaluation (2018) doi: 10.48550/arXiv.1801.07746
7. Asarnow, J. R., Ougrin, D.: Editorial: Suicide and self-harm: Advancing from science to preventing deaths. Journal of child psychology and psychiatry, and allied disciplines, vol. 60, no. 10, pp. 1043—1045 (2019) doi: 10.1111/jcpp.13131
8. Barbieri, F., Camacho-Collados, J., Espinosa-Anke, L., Neves, L.: TweetEval: Unified benchmark and comparative evaluation for tweet classification. Findings of the Association for Computational Linguistics, Empirical Methods in Natural Language Processing, pp. 1644–1650 (2020) doi: 10.18653/v1/2020.findings-emnlp.148
9. Boettler, T., Newsome, P. N., Mondelli, M. U., Maticic, M., Cordero, E., Cornberg, M., Berg, T.: Care of patients with liver disease during the COVID-19 pandemic: EASL-ESCMID position paper. JHEP Reports, vol. 2, no. 3, pp. 100113 (2020) doi: 10.1016/j.jhepr.2020.100113

10. Castillero-Mimenza, O.: Método socrático: Qué es y cómo se aplica en la psicología (2024) psicologiaymente.com/clinica/metodo-socratico
11. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., Schuh, P., Shi, K., Tsvyashchenko, S., Maynez, J., Rao, A., Barnes, P., Tay, Y., Shazeer, N., Prabhakaran, V., Reif, E., et al.: PaLM: Scaling language modeling with pathways. Journal of Machine Learning Research, vol. 24, no. 1, pp. 1–113 (2024)
12. Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W. C., Wang, C. B., Bernardini, S.: The COVID-19 pandemic. Critical reviews in clinical laboratory sciences, vol. 57, no. 6, pp. 365–388 (2020) doi: 10.1080/10408363.2020.1783198
13. Darcy, A., Evers, M.: Woebot health (2023) woebothealth.com/
14. Gerganov, G.: llama.cpp (2023) github.com/ggerganov/llama.cpp
15. Hamilton, J.: Youper (2023) www.youper.ai/
16. Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de-Las-Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van-den-Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Vinyals, O., et al.: Training compute-optimal large language models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, pp. 30016–30030 (2024) doi: 10.48550/arXiv.2203.15556
17. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations, pp. 1–13 (2022) doi: 10.48550/arXiv.2106.09685
18. Inkster, B., Janssen, C. A.: wysa (2023) www.wysa.com/team
19. Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de-las-Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M. A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., Sayed, W. E.: Mistral 7B (2023) doi: 10.48550/ARXIV.2310.06825
20. Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de-las-Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M. A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., et al.: Mixtral of experts (2024) doi: 10.48550/ARXIV.2401.04088
21. Košecká, D., Balco, P.: Use of a communication robot—chatbot in order to reduce the administrative burden and support the digitization of services in the university environment. Developments in Information and Knowledge Management Systems for Business Applications, pp. 597–629 (2023) doi: 10.1007/978-3-031-25695-0_26
22. Lambert, N., Schoelkopf, H., Gokaslan, A., Soldaini, L., Pyatkin, V., Castricato, L.: Self-directed synthetic dialogues and revisions technical report (2024) doi: 10.48550/ARXIV.2407.18421
23. Lee, S. A., Jobe, M. C., Mathis, A. A., Gibbons, J. A.: Incremental validity of coronaphobia: Coronavirus anxiety explains depression, generalized anxiety, and death anxiety. Journal of Anxiety Disorders, vol. 74, pp. 102268 (2020) doi: 10.1016/j.janxdis.2020.102268
24. Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: DailyDialog: A manually labelled multi-turn dialogue dataset. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing, vol. 1, pp. 986–995 (2017) doi: 10.48550/arXiv.1710.03957
25. Manyika, J.: An overview of Bard: An early experiment with generative AI (2023)
26. May, R., Denecke, K.: Security, privacy, and healthcare-related conversational agents: A scoping review. Informatics for Health and Social Care, vol. 47, no. 2, pp. 194–210 (2021) doi: 10.1080/17538157.2021.1983578
27. Mazza, M. G., de-Lorenzo, R., Conte, C., Poletti, S., Vai, B., Bollettini, I., Melloni, E. M. T., Furlan, R., Ciceri, F., Rovere-Querini, P., Benedetti, F.: Anxiety and depression in

COVID-19 survivors: Role of inflammatory and clinical predictors. Brain, Behavior, and Immunity, vol. 89, pp. 594–600 (2020) doi: 10.1016/j.bbi.2020.07.037

28. Megahed, N. A., Ghoneim, E. M.: Antivirus-built environment: Lessons learned from COVID-19 pandemic. Sustainable Cities and Society, vol. 61, pp. 102350 (2020) doi: 10.1016/j.scs.2020.102350
29. Narynov, S., Zhumanov, Z., Gumar, A., Khassanova, M., Omarov, B.: Development of chatbot psychologist applying natural language understanding techniques. In: Proceedings of the 21st International Conference on Control, Automation and Systems, pp. 636–641 (2021) doi: 10.23919/iccas52745.2021.9649825
30. Oblitas, L. A.: Psicoterapias contemporáneas. Cengage Learning Editores (2008)
31. OpenIA: Introducing ChatGPT (2022) chatgpt.com
32. Oppenheimer, C., Silk, J., Jones, N., Butterfield, R.: Neural processing of social rejection and suicide risk. Biological Psychiatry, vol. 87, no. 9, pp. S420 (2020) doi: 10.1016/j.biopsy ch.2020.02.1072
33. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., 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., Leike, J., Lowe, R.: Training language models to follow instructions with human feedback. In: Proceedings of the 36th International Conference on Neural Information Processing Systems, pp. 27730–27744 (2024)
34. Plutchik, R.: A general psychoevolutionary theory of emotion. Theories of Emotion, pp. 3–33 (1980) doi: 10.1016/b978-0-12-558701-3.50007-7
35. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: SQuAD: 100,000+ questions for machine comprehension of text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 2383–2392 (2016) doi: 10.18653/v1/D16-1264
36. Raws, M.: Try tess today! (2023) www.x2ai.com/uprisehealth
37. Rogers, C. R., Farson, R. E.: Active listening. Martino Fine Books (2015)
38. Salhi, I., Guemmat, K. E., Qbadou, M., Mansouri, K.: Towards developing a pocket therapist: An intelligent adaptive psychological support chatbot against mental health disorders in a pandemic situation. Indonesian Journal of Electrical Engineering and Computer Science, vol. 23, no. 2, pp. 1200 (2021) doi: 10.11591/ijeecs.v23.i2.pp1200-1211
39. Shah, K., Kamrai, D., Mekala, H., Mann, B., Desai, K., Patel, R. S.: Focus on mental health during the coronavirus (COVID-19) pandemic: Applying learnings from the past outbreaks. Cureus, vol. 12, no. 3, pp. e7405 (2020) doi: 10.7759/cureus.7405
40. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding (2024) doi: 10.1016/j.neucom.2023.127063
41. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P. B., Hashimoto, T. B.: Alpaca: A strong, replicable instruction-following model (2021) crfm.stanford.edu/20 23/03/13/alpaca.html
42. 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
43. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., et al.: LLaMA 2: Open foundation and fine-tuned chat models (2023) doi: 10.48550/ARXIV.2307.09288
44. Troglio-da-Silva, F. C., Rolim-Neto, M. L.: Psychiatric symptomatology associated with depression, anxiety, distress, and insomnia in health professionals working in patients affected by COVID-19: A systematic review with meta-analysis. Progress in Neuro-Psychopharmacology and Biological Psychiatry, vol. 104, pp. 110057 (2021) doi: 10.1016/j.pnpbp.2020.110057

45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems, pp. 1–15 (2017) doi: 10.48550/ARXIV.1706.03762
46. Wu, M., Waheed, A., Zhang, C., Abdul-Mageed, M., Aji, A. F.: LaMini-LM: A diverse herd of distilled models from large-scale instructions. In: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, vol. 1, pp. 944–964 (2024) doi: 10.48550/arXiv.2304.14402
47. Yáñez, E. A., Reyes, S. V., Gómez, J. F. R., Elizondo, P. V., Gonzalez, A. M., Hernández, A. G.: Implementation of sentiment analysis in chatbots in spanish to detect signs of mental health problems. In: New Perspectives in Software Engineering, vol. 576, pp. 196–214 (2022) doi: 10.1007/978-3-031-20322-0_14

ISSN 1870-4069

Predicting University Student Dropout with Extracurricular Activities Participation Using Machine Learning Models: A Case Study at Tecnológico de Monterrey

Francisco Mestizo, Alberto Orozco, Belén González,
Eunice Santos, Neil Hernandez-Gress

Tecnológico de Monterrey,
Monterrey, Mexico

{A01731549,A00831719,A01625378,A00831991,ngress}@tec.mx

Abstract. This study explores the impact of extracurricular activities on student retention at Tecnológico de Monterrey using data from the Institute for the Future of Education (IFE). The primary purpose of this study is to understand, predict, and prevent university dropout by examining the role of extracurricular activities and other relevant factors. By identifying the key determinants influencing dropout rates, we aim to offer actionable insights that can enhance student retention at a University. The research examines how participation in physical education, cultural activities, and student societies affects dropout rates among undergraduates. The initial analysis did not show a direct link between specific activities and retention, but higher overall engagement was correlated with reduced dropout rates. Machine learning models, including Support Vector Machines (SVM), Decision Trees, and Random Forests, were trained on a balanced dataset, with SVM achieving the highest accuracy at 61% after hyperparameter tuning. The study concludes that increased participation in extracurricular activities improves student retention, emphasizing the need for diverse programs to support student engagement. Also, the statistical analysis reveals that academic performance, financial support, and enrollment status are the primary predictors of student retention. These findings align with existing literature on student dropout, confirming the critical role of these factors. Future research should investigate additional non-academic factors and compare the findings between different institutions.

Keywords: Student dropout, machine learning, extracurricular activities.

1 Introduction

University dropout is a significant issue affecting educational institutions globally, including Mexico. High dropout rates impact students' future opportunities and the overall effectiveness of the educational system. Dropout can lead to financial instability, loss of human capital, and lower socio-economic mobility for individuals.

According to a report by the Organization for Economic Co-operation and Development [6], the dropout rates in higher education are a pressing concern, particularly in developing countries where education is a crucial pathway for social and economic advancement. Addressing this issue is vital for enhancing educational outcomes and ensuring that students can successfully complete their studies [5]. This study aims to explore how well student dropout can be explained by variables not extensively studied in previous research.

Specifically, we investigate the impact of extracurricular activities, along with other potential factors such as student engagement, socio-economic background, and academic support services. By examining these under-explored variables, we aim to provide new insights and contribute to the existing body of knowledge on student retention strategies. At Tecnológico de Monterrey, a variety of “LiFE” [1] courses and student groups are offered to enrich the student experience. These include sports teams, cultural clubs, leadership programs, and wellness activities.

These initiatives are designed to i) develop softskills and, ii) foster a sense of community, promote personal development, and enhance students’ overall well-being. Understanding the role these activities play in student retention is crucial, as they are integral to the Tec’s educational philosophy and mission to develop well-rounded individuals. The primary purpose of this study is to understand, predict, and prevent university dropout by examining the role of extracurricular activities and other pertinent factors. By identifying the key determinants influencing dropout rates, we aim to offer actionable insights that can enhance student retention at Tecnológico de Monterrey[2]:

– Research Question: Can the extracurricular activities offered by Tecnológico de Monterrey be a significant factor in preventing students from dropping out?

– Hypothesis: Extracurricular activities provided by Tecnológico de Monterrey play a relevant role in reducing student dropout rates.

This paper is organized as follows: Section 2 describes the State of the Art in Academic Performance and Student Dropout as well as Predictive Models. Section 3 describes the Database that has been developed [3] describing indicators and variables. Section 4 explains the Results of applying different Machine Learning methodologies to the Database. Finally Section 6 Concludes and discusses the future work.

2 State of the Art

2.1 University Dropout

University dropout has been extensively studied across various contexts, with numerous factors identified as key determinants. Among these factors, academic performance, socioeconomic status, extracurricular participation, and family support stand out as significant contributors.

[1] studyinmexico.tec.mx/es/life
[2] www.tec.mx

Academic Performance: Academic struggles, particularly in the first year, have been consistently linked to higher dropout rates. Students who fail to meet academic standards or experience difficulty adjusting to the academic rigors of university life are more likely to leave their studies prematurely. Rooij, Hansen, and Grift(2018) [12] found that first-year academic performance was a strong predictor of student retention in universities. Their research highlighted that early academic success is crucial for maintaining student engagement and persistence in higher education.

Socioeconomic Status: Socioeconomic challenges, including financial instability and the need to balance work with studies, are critical factors influencing dropout rates. Students from lower-income families often face additional pressures that can impact their ability to remain enrolled. Financial aid and scholarship programs have been found to play a crucial role in mitigating these challenges and improving retention rates. Aina, Baici, [1] demonstrated that socioeconomic status significantly affects student retention and graduation rates, emphasizing the need for robust financial support systems to aid disadvantaged students.

Extracurricular Participation: Engagement in extracurricular activities, such as sports, cultural clubs, and leadership programs, has been positively associated with student retention. These activities help students build a sense of community and belonging, which can be vital for their overall well-being and academic success. [8] highlighted the importance of extracurricular engagement in enhancing student retention and success. Their study showed that students who participate in extracurricular activities are more likely to stay enrolled and achieve academic success.

Family Support: The level of family support, including parents' educational background and involvement in their children's education, significantly affects student persistence. Research indicates that students whose parents have higher educational attainment levels are more likely to complete their studies. Werfhorst, [7] explored the impact of parental education on student persistence in higher education, finding that higher parental education levels correlate with increased student retention rates.

2.2 Predictive Models

Machine learning models have gained prominence in general and more specific in predicting student dropout due to their ability to handle large datasets and identifying complex patterns. Several models have been utilized to predict student retention, each with its strengths and limitations. Decision Trees: Decision trees are a non-parametric supervised learning method used for classification and regression. They provide a visual representation of decision rules and are easy to interpret.

However, decision trees can be prone to over-fitting, especially with small datasets. [2] applied decision tree analysis to identify at-risk students and found it effective in highlighting key predictors of dropout. Their study underscored the importance of decision trees in educational data mining due to their simplicity and interpretability. Random Forests: Random forests, an ensemble learning method, improve upon decision trees by constructing multiple trees during training and outputting the class that is the mode of the classes of individual trees.

This method reduces overfitting and increases accuracy, making it a robust choice for predictive modeling. [10] found that random forests provided superior performance in predicting student dropout compared to other machine learning methods. Their research highlighted the robustness and accuracy of random forests in educational settings. Support Vector Machines (SVM): SVMs are powerful for classification tasks, particularly when the data is high-dimensional. They work by finding the hyperplane that best separates the data into classes. SVMs are effective in capturing complex relationships between variables, though they require careful tuning of parameters.

In one study [9], the main predictors that are used to estimate student dropout are: Regime, Application grade, Internet, Car, Educational level of Parents, Occupation, Private insurance, Part-Time Job, Desktop Computer, Laptop. The results of this investigation lead to an accuracy approximate to 76.8%. [9] Another study (biblio) related with student dropout prediction using Machine Learning focuses on independent variables that are more related to student metrics about their own performance in school, mood during days of class and a few personal features: Gender, Former education level, Application priority, Degree program, Mother language, Start Semester, Age at enrollment, Credits, GPA, Failed Courses, exchange days, Moodle count, Moodle trent.

This study obtained results that were approximate to an accuracy of 76% determining the output if the student graduated or not[11]. While these researches focus on specific attributes to determine student dropout, like the student's performance in school, mood, socioeconomic and some other personal characteristics, this research opted for using the same approach but with a different focus. Most of the variables that were considered are related to the student's involvement in extracurricular activities, so we could determine dropout based on the influence of these activities.

3 Experimental Development

In order to have a better understanding of the proposed methodology for solving the designated challenge, it is necessary to establish some basic concepts. For this, it is essential to talk about Artificial Intelligence itself, more specifically, Machine Learning. This is described as the use of computational methods to be trained from data without the need of managing manual control in the applied procedure. It is to say that artificial intelligence models are being used to solve problems of almost any nature automatically. [4]

For this research, multiple options are considered in order to find the best model possible with the best metrics. In Machine Learning, exist multiple models that can provide good estimations for student dropout in college, like the Support Vector Machine, Neural Networks, Decision Tree and Random Forests to name a few related to supervised learning, also there are models for clustering like K-Means or DBSCAN used in unsupervised learning.

Based on the research performed, there were found interesting methodologies applied for very similar problems that were carried out with Machine Learning. In one study there was mentioned that supervised learning has the most accurate results with three main models that have proven to be the most outstanding ones referring to achieving a robust classification performance and distinguished accurate results [13].

The first one is the Support Vector Machines model, this algorithm is capable of performing classification tasks such as binary and multi-class classification, as well as regression problems. The objective of this model is finding the hyperplane that better divides the data classes. Some of the advantages of applying decision trees are:

- Decision trees are easy to understand, interpret and visualize.
- Can handle categorical as well as numeric variables without the need of complex pre-processing.
- It doesn't require data normalization or feature standardization.
- Captures non.linear relations between features and tags.

4 Results

The data that supports the findings of this study are available from the Institute for the Future of Education (IFE)'s Educational Innovation collection of the Tecnológico de Monterrey's Research Data Hub but restrictions apply to the availability of these data, which were used under a signed Terms of Use document for the current study, and so are not publicly available. However, data is available from the IFE Data Hub upon reasonable request at https://doi.org/10.57687/FK2/PWJRSJ (accessed on 11 April 2024). The used database has information from Tec de Monterrey's high school and undergraduate students [3]. The dataset has information related to academic performance, participation in extracurricular activities inside the school and information from the parents of the students. To reduce the scope for this research, it is only focused in the undergraduate students. The description of each category will be given below:

1. Physical Education: Students enrolled in one or more physical activities. One in this column represent that the student is enrolled and zero means they are not enrolled.
2. Cultural Diffusion: Students enrolled in one or more cultural activities. One in this column represent that the student is enrolled and zero means they are not enrolled.
3. Student Society: Students actively involved in one or more student groups. One in this column represent that the student is involved and zero means they are not involved.
4. Total LiFE Activities: LiFE activities are the extracurricular activities offered by Tec de Monterrey. This column indicates in how many activities the student is enrolled.
5. Athletic Sports: Students that are part of any team that represent the university in different sports competition.
6. Art Culture: Students that are part of any team that represent the university in different cultural activities, such as theater and dance.
7. Student Society Leadership: Students that are part of the board of any student society.
8. LiFE Work Mentoring: Student that are part of programs like Peer Mentor, where they help freshmen and sophomores.

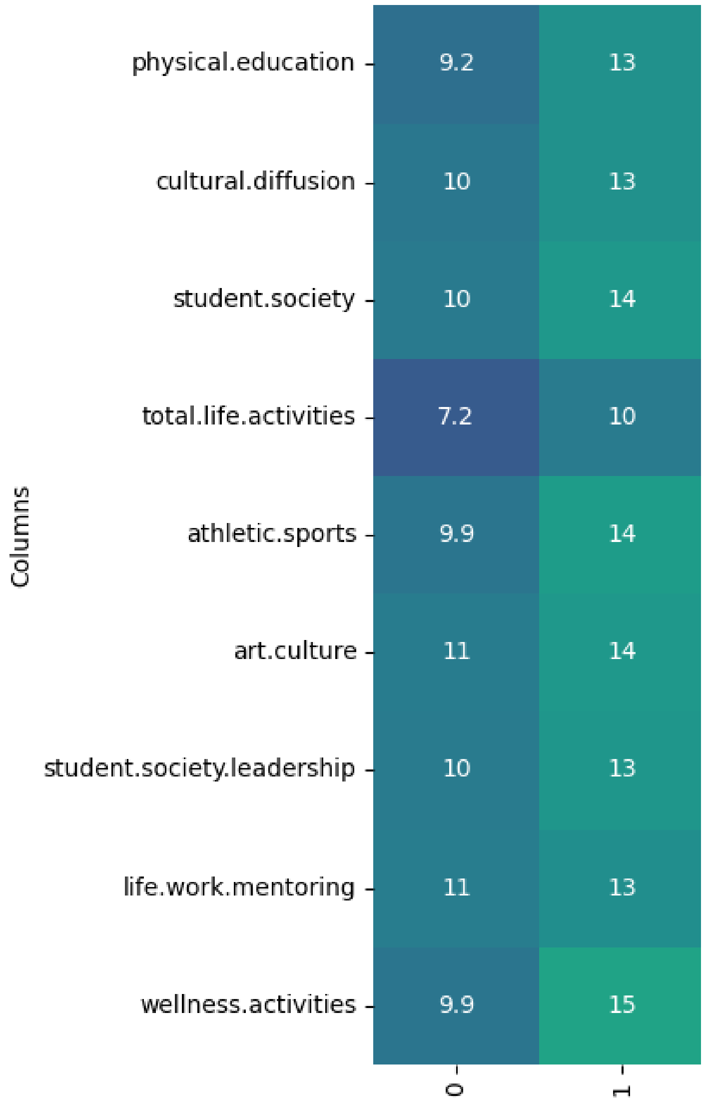


Fig. 1. Ratio of school permanence by activity.

9. Wellness Activities: Students enrolled in well-being activities, like meditation.
10. Foreign: Indicator to identify if the student is a foreigner (Yes: Foreigner), if the student's birthplace is different from the location of the school campus (Yes: National) or belongs to the same location (Local).
11. PNA: Previous level score (Average).

The value to be predicted is represented in the database as Retention. A value of one means the student is still studying, while a value of zero means they left school.

4.1 Data Cleaning

To start using the data it has to be cleaned. Overall, it is a well-structured database, as it does not have any null value for any column. Also, most of the columns are self-described so it is easy to understand what they are measuring just by reading the name. Some of the columns contains data as “No information” or “Does not apply”. For this data to be used for the models, that values where changed for 0. For the categorical columns as foreign, one hot encoding was performed.

4.2 Data Analysis

After the columns where selected, a correlation analysis with retention was conducted to understand the impact these variables have in retention. All the variables had a correlation coefficient of less than 5%, so the variables do not have direct correlation with retention. It was analysed how much impact each category had by itself. This was done by comparing all the students that where on a physical activity and dropped with all the students that were not in a physical activity compared to the students that were on a physical activity and stayed in school. With that result, a ratio was calculated.

This was done with each variable. The result showed a biggest ratio of participation in any activity for students that did not quit school. The difference in these ratios is on average of three points as shown in figure 1. For total life activities, a curious pattern is shown, where having one extra LiFE activity increments by average three points the ratio. This means that the more LiFE activities an student has, the least probable it is that they quit school. So, even though this categories did not show direct correlation with retention by themselves, their presence appears to be important.

4.3 Data Sampling

The dataset contains information from all the students from generation 2014 through 2020. Just for this generations, there are 77,517 students in the database. From those, 70,704 stayed in school while the other 6,813 quit school. It is reassuring to see that the number of dropouts is much lower than the students that stayed, but it represents a problem for training a machine learning model [13]. The desired dataset for training a machine learning model is to have balanced classes, this means to have similar amount of data for each class. There are several methods to accomplish this, like applying data augmentation by generating random data from dropouts to balance both classes. Another approach, which is the one taken on this research, is to take all dropout registers and take randomly the same amount for students that stayed on school. This generates a final dataset with 50% - 50% distribution of the classes.

4.4 Model Training and Selection

According to Villar [13], the models that have shown best results for predicting student dropout are Support Vector Machines (SVC), Decision Trees and Random Forest. No hyperparameters will be used for this stage. The models will be trained using the default values for their hyperparameters.

Those three models were trained with the same balanced dataset to compare the performance of each one based on accuracy. Then, the best one will be selected for a refinement of its hyperparameters. SVM Model: 0.58, Decision Tree: .54 and Random Foret 0.54 The results for each model can be seen in the tables 1 through 3. On average, the results of accuracy for the models are 56%, with the best model being SVM with 59% of accuracy. The results are not optimal, but they may be improved by refining the model.

4.5 Model Refinement

Some of the hyperparameters that can be defined for a SVM model are:

- C (Regularization parameter): Controls the trade-off between achieving a low error on the training data and minimizing the complexity of the model (which helps in avoiding over-fitting).
- Gamma: Defines how far the influence of a single training example reaches.
- Kernel: Specifies the type of kernel to be used in the SVM algorithm. Linear means a linear boundary is used to separate the data, while Radial Basis Function (RBF) uses non-linear decision boundaries.

For the optimization, different combinations of those hyperparameters where tested. For C, the values where 0.1, 1 and 10; for Gamma, 0.1 1 and 10; and for kernel, linear and RBF. The values selected where arbitrary, just to test different states of the model. After being trained with the same dataset, the best model had the values. The results for accuracy are 0.58. The result do not show great improvement for the model. This could mean that this values cannot predict more than 60% of the data. As this is not a great result, it is quite acceptable for variables that were not that correlated to the predicted value.

4.6 Model Improvement

To test how much the model could be improved, the same steps where taken but using more categories. To select the new classes a Principal component analysis (PCA) and correlation analysis was done.

Principal Components Analysis. Principal Component Analysis (PCA) was performed to reduce the dimensionality of the data and identify the main components contributing to variance. The first two principal components explain approximately 32.2% of the total variance, indicating that while these components capture a significant portion of the data's variability, other factors also contribute to the overall dropout rates. These findings indicate that academic performance (Average First Period Grade) is the most significant predictor of student retention. This aligns with the general understanding that students with better academic performance are less likely to drop out. Financial support (Scholarship Percentage and Total Scholarship Loan) and enrollment status (FTE) also play crucial roles, suggesting that students who receive financial aid and are fully enrolled are more likely to persist in their studies.

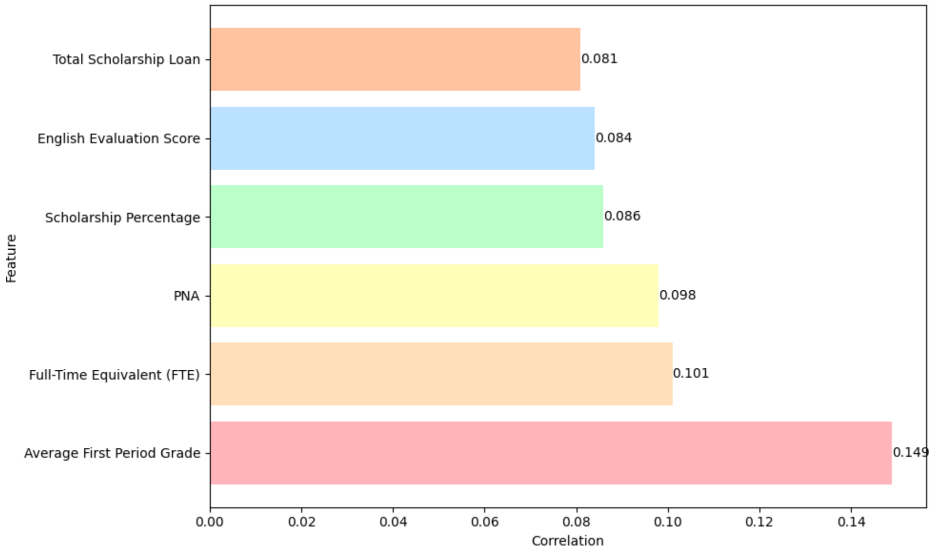


Fig. 2. Top correlated features with retention.

The PNA and English Evaluation Score reflect engagement and proficiency, further underlining the importance of academic and support structures.

Correlation Analysis. The correlation between each feature and the target variable (retention) was computed. The top six correlated features are shown in Figure 2. The correlation analysis supports the findings from the feature importance analysis. High correlations between retention and features like Average First Period Grade and Scholarship Percentage emphasize the critical role of academic performance and financial support in preventing dropout.

4.7 Model Re-training

After the mentioned analysis, the added categories were:

1. English Evaluation: Level of English obtained from a standardized English proficiency test.
2. Tec no Tec: Indicator that denotes if the student comes from a school that belongs to Tecnológico de Monterrey.
3. Parents Exatec: Indicator that denotes if either one of the two parents is exatec (was a student of Tecnológico de Monterrey).
4. General Math Eval: Mathematics grade from the admission test or from the school of origin.

Three of this categories are related to academic performance, while the other is centered in the school background of the family.

For this training, the same data sampling and cleaning methods were used. The only difference was for the General Math Eval column. For the values of “No information” and “Does not apply” zeros were not added as for the other columns. That could cause problems for the predictions because 0 have a different meaning in this column. That’s why the mean of the other values of the column was obtained and then changed for the registers that did not had a numerical value. With the addition of these the variables we can see a slightly increase in accuracy, but it doesn’t seem to be too significant. The best model was SVM again, with an accuracy of 61%. The result for other models are: Decision Trees: 0.55, Random Forest: 0.56.

5 Analysis

The statistical analysis reveals that academic performance, financial support, and enrollment status are the primary predictors of student retention. These findings align with existing literature on student dropout, confirming the critical role of these factors. But, extracurricular activities, although not among the top predictors, show a noticeable impact on retention, as shown in this research. This supports the hypothesis that these activities foster engagement and a sense of belonging, which are essential for student persistence. The findings are noticeable by the dataset from Tecnológico de Monterrey, which includes detailed information on academic performance and financial support but also from extracurricular involvement.

The use of advanced machine learning models, particularly Support Vector Machines, allowed uncovering these insights with high accuracy. The study demonstrates the effectiveness of using machine learning models to predict student dropout and identify key factors influencing retention, even thought the variables had no correlation at all. The findings suggest that institutions should focus on enhancing academic support, providing financial aid, and promoting extracurricular activities to improve student retention. Future research could explore the specific types of extracurricular activities that have the most significant impact and develop tailored interventions to support at-risk students.

5.1 Student Retention through Extracurricular Programs

Based on the results, its proposed that other institutions adopt and expand extracurricular activities similar to those offered at Tecnológico de Monterrey. Programs like LIFE, which integrate leadership, innovation, and entrepreneurial training with traditional extracurricular activities, could be particularly beneficial. These programs not only enhance student engagement but also build skills that contribute to academic and professional success. Below are examples of extracurricular activities that institutions could adopt:

– Leadership Programs: Workshops and seminars that develop leadership skills and provide opportunities for students to take on leadership roles.

– Innovation and Entrepreneurship: Incubators and hackathons that encourage creative problem-solving and entrepreneurial thinking.

– Sports and Fitness: Comprehensive sports programs that promote physical health and team-building skills.

– Cultural and Arts Programs: Activities that celebrate cultural diversity and foster creative expression.

– Community Service: Volunteer opportunities that help students develop empathy and a sense of social responsibility.

– Professional Development: Career counseling, internships, and networking events that prepare students for post-graduation success.

6 Conclusion and Future Work

Support Vector Machines, as seen in other bibliography, resulted in one of the most effective machine learning models to predict student dropout. The use of variables such as students' extracurricular performance, although not strongly correlated with their retention, proved to be variables that could predict up to 61% of the cases of students who remained in or left school. From this research, it is encouraged to use machine learning models in other variables not related to academic performance, such as extracurricular performance, geographical context, health, family, technological development or emotional state. Also, it could be interesting to do this same analysis in other schools near Tecnológico de Monterrey to see if the obtained results are similar or do not.

Acknowledgments. The authors would like to thank Tecnológico de Monterrey and the Living Lab and Data Hub of the Institute for the Future of Education for the data provided for this research.

References

1. Aina, C., Baici, E., Casalone, G., Pastore, F.: The determinants of university dropout: A review of the socio-economic literature. Socio-Economic Planning Sciences, vol. 79, pp. 101102 (2022) doi: 10.1016/j.seps.2021.101102
2. Albreiki, B., Habuza, T., Zaki, N.: Extracting topological features to identify at-risk students using machine learning and graph convolutional network models. International Journal of Educational Technology in Higher Education, vol. 20, no. 1 (2023) doi: 10.1186/s41239-023-00389-3
3. Alvarado-Uribe, J., Mejía-Almada, P., Masetto-Herrera, A. L., Molontay, R., Hilliger, I., Hegde, V., Montemayor-Gallegos, J. E., Ramírez-Díaz, R. A., Ceballos, H. G.: Student dataset from Tecnológico de Monterrey in Mexico to predict dropout in higher education. Data, vol. 7, no. 9, pp. 119 (2022) doi: 10.3390/data7090119
4. Bhasin, H.: Machine learning for beginners: Learn to build machine learning systems using python. BPB Publications (2020)
5. Cannistrà, M., Masci, C., Ieva, F., Agasisti, T., Paganoni, A. M.: Early-predicting dropout of university students: An application of innovative multilevel machine learning and statistical techniques. Studies in Higher Education, vol. 47, no. 9, pp. 1935–1956 (2021) doi: 10.1080/03075079.2021.2018415

6. Canton, H.: The Europa directory of international organizations. Routledge (2021) doi: 10.4324/9781003179900
7. Forster, A. G., van-de-Werfhorst, H. G.: Navigating institutions: Parents' knowledge of the educational system and students' success in education. European Sociological Review, vol. 36, no. 1, pp. 48–64 (2019) doi: 10.1093/esr/jcz049
8. King, A. E., McQuarrie, F. A., Brigham, S. M.: Exploring the relationship between student success and participation in extracurricular activities. SCHOLE: A Journal of Leisure Studies and Recreation Education, vol. 36, no. 1–2, pp. 42–58 (2020) doi: 10.1080/1937156x.2020.1760751
9. Sandoval-Palis, I., Naranjo, D., Vidal, J., Gilar-Corbi, R.: Early dropout prediction model: A case study of university leveling course students. Sustainability, vol. 12, no. 22, pp. 9314 (2020) doi: 10.3390/su12229314
10. Utari, M., Warsito, B., Kusumaningrum, R.: Implementation of data mining for drop-out prediction using random forest method. In: Proceedings of the 8th International Conference on Information and Communication Technology, pp. 1–5 (2020) doi: 10.1109/ICoICT49345.2020.9166276
11. Vaarma, M., Li, H.: Predicting student dropouts with machine learning: An empirical study in finnish higher education. Technology in Society, vol. 76, pp. 102474 (2024) doi: 10.1016/j.techsoc.2024.102474
12. van-Rooij, E. C. M., Jansen, E. P. W. A., van-de-Grift, W. J. C. M.: First-year university students' academic success: The importance of academic adjustment. European Journal of Psychology of Education, vol. 33, pp. 749–767 (2017) doi: 10.1007/s10212-017-0347-8
13. Villar, A., de-Andrade, C. R. V.: Supervised machine learning algorithms for predicting student dropout and academic success: A comparative study. Discover Artificial Intelligence, vol. 4, no. 1 (2024) doi: 10.1007/s44163-023-00079-z

ISSN 1870-4069

US Airlines Twitter Opinion Analysis: Classifying Positive or Negative Comments

Raul Jimenez-Cruz, Galo Ruiz-Soto,
Miguel Gonzalez-Mendoza

Tecnologico de Monterrey,
School of Engineering and Sciences,
Mexico

{r.jimenez.c, a01799399, mgonza}@tec.mx

Abstract. This study investigates the classification of sentiment in tweets related to airlines, aiming to determine whether opinions are positive or negative. The dataset includes features such as the airline mentioned, sentiment ranking, geolocation, and the sentiment label. Six classifiers were evaluated for their effectiveness in sentiment classification. The preprocessing phase involved lemmatization, the removal of stopwords to clean the text data, and the generation of bigrams to mitigate the sparsity of the sparse matrix. Given the dataset's imbalance with an Imbalance Ratio (IR) of 1.68, the balanced accuracy metric was employed to ensure a fair assessment of classifier performance. The classifiers' outputs were geographically mapped to provide a visual representation of sentiment distribution, facilitating a more tangible analysis of the results. Among the classifiers, Logistic Regression achieved the highest accuracy (0.7281), while Multinomial Naive Bayes obtained the best balanced accuracy (0.7920). This study demonstrates the importance of robust preprocessing and the selection of appropriate evaluation metrics in handling imbalanced datasets, contributing valuable insights into the performance of different classifiers in sentiment analysis tasks within the domain of natural language processing and machine learning.

Keywords: Sentiment analysis, airline, classification, algorithm, prediction, machine learning, text analysis.

1 Introduction

In February 2015, travelers provided opinions about their experiences using six different US airlines: American, United, Southwest, Delta, Virgin America, and US Airways. These opinions were classified as positive, negative, or neutral. Airlines recognize that detecting and understanding customer emotions is crucial for providing a superior customer experience. Satisfied customers are more likely to repurchase tickets from the same airline, as emotions can significantly influence future buying decisions. Identifying emotions promptly can help airlines adjust their services to improve customer moods, thereby reducing churn rates. Anticipating user emotions is key for securing upskilling opportunities and retaining customers.

In this study, neutral sentiments were merged with positive sentiments to address the challenge of class imbalance inherent in the dataset. This approach simplifies the classification task, making it more manageable and allowing for more robust model performance, particularly when using balanced accuracy as a metric. The conversion to a two-class problem aligns with the research objective of assessing overall customer satisfaction, where neutral opinions are often considered closer to positive than negative in this context. This paper aims to predict the sentiments of US airline users by analyzing their emotions expressed on Twitter.

The study will involve a comprehensive analysis of the data to identify any issues that could impact the results. Various classification models will be employed, including K Nearest Neighbors (KNN) with 1 and 3 neighbors, Multinomial Naïve Bayes, Random Forest, Gradient Boosting, and Logistic Regression, to predict the sentiments. The performance of these models will be assessed using accuracy metrics and confusion matrices. In particular, the balanced accuracy metric will be used to address potential biases given the imbalanced nature of the dataset. As in [10] have argued, accuracy should not be taken as an absolute measure. Thus, the study will explore alternative performance measures to ensure robust model evaluation.

2 Methodology

2.1 Dataset

The dataset is publicly available on Kaggle [8]. The dataset comprises 15 columns: 'tweet_id', 'airline_sentiment', 'negative_reason', 'airline', 'airline_sentiment_ confidence', 'retweet_count', 'text', 'negativereason_confidence', 'airline_ sentiment_gold', 'name', 'negativereason_gold', 'tweet_coord', 'tweet_location', 'user_timezone', and 'tweet_created'.

However, most of these columns are irrelevant for the current task. For instance, 'tweet_id', 'name', 'tweet_coord', and 'tweet_created' do not contribute to the sentiment classification task. The dataset contains opinions from Twitter users about US airlines, categorized into three classes: positive, negative, and neutral, as indicated in the 'airline_sentiment' column, which is the target variable for prediction. Due to the irrelevance of most columns to the problem, only the 'airline_sentiment' and 'text' columns were initially retained.

However, the neutral class was merged with the positive class to address class imbalance, reducing bias observed in initial experiments where the neutral class skewed towards positive. This adjustment resulted in a class imbalance ratio of 1.68. In addition to the 'text' column, the 'airline_sentiment_confidence' and airline columns were included as features.

The 'airline' column underwent label encoding to convert textual data into numerical format. The dataset contains 14,640 messages: 9,178 negative, 3,099 neutral, and 2,363 positive before merging classes. After combining the neutral and positive classes, the distribution is more balanced but still poses an imbalance challenge. There are no missing values in the columns. The messages are distributed across six airlines: United (3,822 messages), US Airways (2,913 messages), American (2,759

Table 1. Messages divided by airline.

Airline	Negative	Neutral	Positive
American	1,960	463	336
Delta	955	723	544
Southwest	1,186	664	570
US Airways	2,263	381	269
United	2,633	697	492
Virgin America	181	171	152

messages), Southwest (2,420 messages), Delta (2,222 messages), and Virgin America (504 messages). A breakdown of negative messages reveals 2,910 concerning customer service issues, 1,665 about late flights, 847 regarding canceled flights, 724 about lost luggage, and 1,190 messages that are ambiguous. Table 1 provides a detailed breakdown of the dataset. The preparation of the data ensures that the features selected are pertinent for the classification task and addressing class imbalance through the combination of neutral and positive classes.

2.2 Preprocess

The first step in the methodology was to clean the text field of the dataset. This involved several preprocessing actions to ensure the text data was suitable for analysis. Initially, a dictionary of contractions was created and used to expand contractions into their full forms. Following this, the text field was cleaned by removing numbers and punctuation. Further preprocessing included lemmatization and the elimination of stopwords, with all text converted to lowercase to ensure uniformity. Lemmatization is the process of reducing words to their base or root form, which helps in normalizing the text and reducing the dimensionality of the data [9].

Stopwords are common words that are typically removed from text data because they do not contribute significant meaning and can lead to noise in the analysis [12]. Once the tweets were cleaned, a new DataFrame was created containing the ‘text’, ‘airline_sentiment_confidence’, and ‘airline’ columns. The ‘airline’ column was subjected to label encoding, converting the categorical text data into numerical labels. The ‘airline_sentiment’ column, which served as the target variable, was similarly label encoded. The next crucial step was vectorizing the ’text’ column.

Term Frequency-Inverse Document Frequency (TF-IDF) was employed for this purpose. TF-IDF is a statistical measure used to evaluate the importance of a word in a document relative to a collection of documents [11]. An important modification was made: bigrams were utilized to capture more context and reduce the feature space. A bigram is a sequence of two adjacent words in a text, which helps in understanding the context better than individual words [7]. Additionally, words appearing fewer than three times were excluded to further streamline the feature set. The resulting sparse matrix from the TF-IDF vectorization was then converted into a DataFrame.

Table 2. The top 10 most important features determined by random forest classifier.

Feature	Importance
airline sentiment confidence	0.551708
airline	0.152976
flight cancelled flightled	0.009168
cancelled flightled flight	0.007575
cancelled flight flight	0.004096
flight booking problem	0.003839
flight cancelled flighted	0.003822
reflight booking problem	0.003039
worst customer service	0.002816
great customer service	0.002627

This DataFrame, now containing the TF-IDF features, was merged with the 'airline_sentiment_confidence' and label-encoded 'airline' columns, consolidating all relevant features for the classification task. This preprocessing pipeline aimed to prepare the data for effective sentiment classification by enhancing the quality and relevance of the features. After preprocessing, the next step was to analyze the new distribution of classes and the data, ensuring the changes adequately addressed the class imbalance and prepared the dataset for model training and evaluation. By carefully cleaning the text data, encoding categorical variables, and using TF-IDF with bigrams, the preprocessing aimed to create a robust feature set for the subsequent classification models. These steps were essential to mitigate noise and imbalance, thus improving the accuracy and reliability of the sentiment predictions.

2.3 Analysis of Preprocessed Patterns

The first step in the analysis involved visualizing the distribution of the classes to observe the imbalance. The negative class contains 9,178 patterns, while the combined positive class has 5,462 patterns. Next, a Random Forest classifier was trained on the entire dataset to determine which features were most important. The top 10 most important features across the dataset are shown below: Table 2 above displays the features and their respective importance scores as determined by the Random Forest classifier.

The most important feature is 'airline_sentiment_confidence', followed by 'airline'. The remaining features, although having lower importance scores, indicate specific terms related to customer issues such as flight cancellations and booking problems. This analysis helps in understanding which features contribute most significantly to the sentiment classification task. The analysis of the most representative words for each class revealed distinct patterns.

For the positive class, terms such as ”fleet fleek,” ”customer service,” and ”thank much” were predominant, highlighting a focus on positive customer experiences and gratitude. In contrast, the negative class was dominated by terms related to service issues and disruptions, such as ”customer service,” ”cancelled flightled,” and ”late flight.” These findings suggest that positive tweets often emphasize appreciation and specific positive interactions, while negative tweets predominantly address complaints about service failures and delays. This lexical analysis provides valuable insights into the differing nature of customer feedback based on sentiment, which can be crucial for tailoring responses and improving service quality.

2.4 Classification

In this section, six different classifiers were implemented and evaluated: KNN 1, KNN 3, Multinomial Naive Bayes, Random Forest, Gradient Boosting, and Logistic Regression. Aggarwal and Zhai [1] provide a comprehensive survey of text classification algorithms, highlighting the effectiveness of machine learning techniques in text mining. Each classifier was subjected to k-fold cross-validation with k=10 to ensure robust performance evaluation. Below is a brief description of each classifier along with relevant references.

1. **K-Nearest Neighbors (KNN):** KNN 1 and KNN 3: KNN is a non-parametric method used for classification and regression. In KNN, the input consists of the k closest training examples in the feature space. KNN 1 uses the closest neighbor, while KNN 3 uses the three closest neighbors [3].
2. **Multinomial Naive Bayes:** This classifier is based on Bayes’ theorem and is particularly suited for classification with discrete features. The multinomial variant is specifically useful for text classification where word frequencies are used as features [6].
3. **Random Forest:** This ensemble learning method constructs multiple decision trees during training and outputs the class that is the mode of the classes of the individual trees. It is known for its robustness and ability to handle overfitting [2].
4. **Gradient Boosting:** Gradient Boosting builds models sequentially, with each new model attempting to correct the errors of the previous models. It combines the predictions of multiple base estimators to improve robustness [5].
5. **Logistic Regression:** This linear model estimates the probability that an instance belongs to a particular class. It is particularly useful for binary classification problems but can be extended to multiclass problems [4].

The classifiers were configured with specific hyperparameters to optimize their performance. For Multinomial Naive Bayes, alpha was set to 0.001 with fit_prior was True. Random Forest used criterion was gini, number of estimators were 100. Logistic Regression was configured with max_iter in 300, solver was liblinear and class_weight was balanced. Given the dataset’s class imbalance, where neutral sentiments significantly outnumber negative ones, we opted to merge neutral with positive sentiments.

Table 3. Performance measures.

Model	Accuracy	Sensitivity (Recall)	Specificity	Balanced Accuracy
KNN 1	0.5210	0.5210	0.3706	0.4458
KNN 3	0.4596	0.4596	0.2294	0.3445
Multinomial Naive Bayes	0.7001	0.7001	0.8839	0.7920
Random Forest	0.7152	0.7152	0.7712	0.6966
Gradient Boosting	0.7050	0.7050	0.8590	0.6526
Logistic Regression	0.7281	0.7281	0.7636	0.7158

This decision was guided by the objective of simplifying the classification task and improving model performance. Balanced accuracy was chosen as the primary metric to ensure that the classifier's performance was fairly evaluated across both classes, particularly in an imbalanced setting.

3 Results

Finally, the performance metrics, including accuracy, balanced accuracy, sensitivity (recall), and specificity, were recorded for each classifier. The results are summarized below: The table 3 above shows the average performance metrics obtained through 10-fold cross-validation for each classifier. Logistic Regression achieved the highest average accuracy (0.7281) but Multinomial Naive Bayes achieved the best balanced accuracy (0.7920), indicating its robustness in handling the class imbalance. Random Forest and Gradient Boosting also performed well, with Gradient Boosting achieving a balanced accuracy of 0.8590. These results provide insights into the effectiveness of different classifiers in predicting user sentiments based on Twitter data.

4 Analysis and Discussion

The results obtained from the classification models provide insightful observations about the performance and applicability of various machine learning algorithms in sentiment analysis of airline tweets. The models implemented were KNN ($k = 1$ and $k = 3$), Multinomial Naive Bayes, Random Forest, Gradient Boosting, and Logistic Regression.

Feature Importance: The feature importance analysis using Random Forest revealed that airline_sentiment_confidence was the most critical feature, followed by airline. Bigram features such as 'flight cancelled flightled' and 'customer service' also ranked highly, indicating their relevance in sentiment classification.

Word Representation Analysis: The representative words for each class highlighted distinct patterns. For the positive class, phrases like "great flight" and "thank much" were prominent, reflecting satisfaction and gratitude. In contrast, the negative class was dominated by terms like "customer service", "cancelled flight", and "late flight", pointing to common issues faced by passengers. To further enrich our analysis, we leveraged the 'tweet_coord' column to map the sentiment geographically.

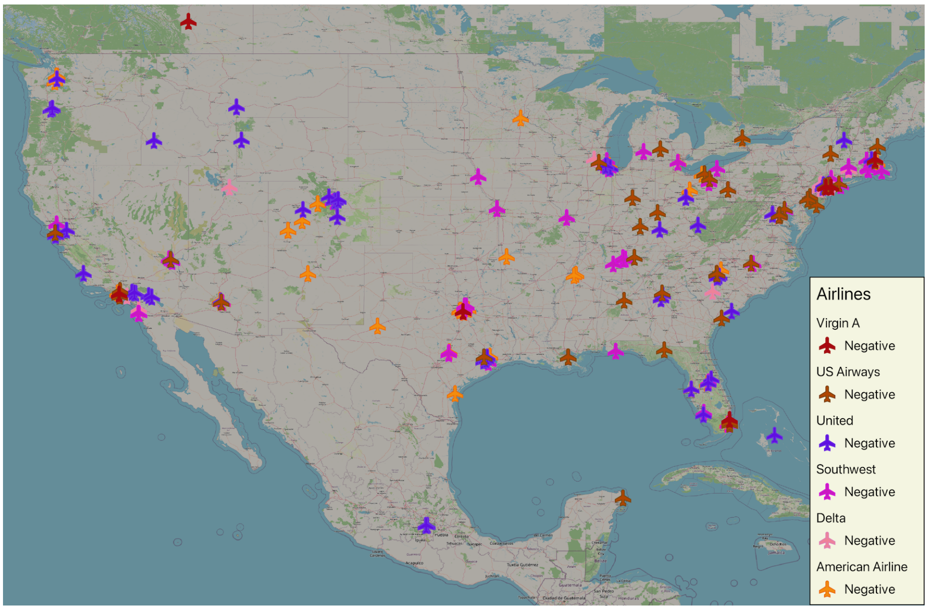


Fig. 1. Positive distribution by airline.

This visualization provides deeper insights into regional sentiment trends and highlights areas with higher concentrations of negative or positive sentiments.

4.1 Negative Sentiments Fig. 1

1. Concentrations of negative sentiments are evident in large metropolitan areas, particularly on the East and West Coasts.
2. Virgin America shows a significant number of negative tweets in Los Angeles and New York, while US Airways has more negative sentiments spread across the Midwest and Northeast.
3. There seems to be a stronger presence of negative sentiments in regions with heavy air traffic.

4.2 Positive Sentiments Fig. 2

1. Positive sentiments are more geographically dispersed, with a noticeable presence in the Midwest and along the East Coast.
2. Virgin America and Southwest Airlines receive positive sentiments in key cities like San Francisco and Dallas.
3. The spatial distribution of positive tweets suggests that positive experiences are more spread out, possibly reflecting regional differences in service quality or customer expectations.

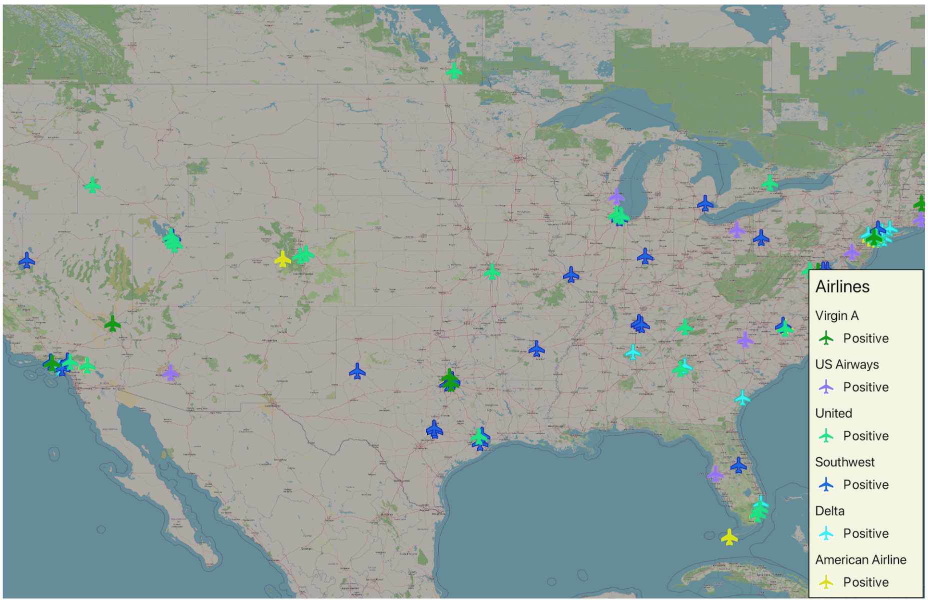


Fig. 2. Positive distribution by airline.

This analysis indicates that airlines could focus on specific regions to improve customer satisfaction, particularly where negative sentiments are concentrated. The geographical patterns could help target marketing efforts or service improvements in those areas.

5 Conclusions

This study demonstrates the effectiveness of various machine learning models in classifying sentiments expressed in airline tweets. Multinomial Naive Bayes emerged as the best-performing model, closely followed by Gradient Boosting. The incorporation of feature engineering techniques, such as bigrams and TF-IDF vectorization, significantly contributed to the models' performance. Addressing the class imbalance through techniques like balanced class weighting combining the neutral and positive class and using balanced accuracy as a performance metric provided a more nuanced understanding of the models' capabilities. Future research could explore several avenues to enhance the current work:

1. **Incorporating Deep Learning Models:** Utilizing advanced deep learning architectures like LSTM and BERT for sentiment analysis might improve accuracy and capture more complex patterns in the text data.
2. **Enhancing Geospatial Analysis:** Integrating detailed geospatial analysis by mapping tweets to specific locations can provide deeper insights into regional sentiment trends and potentially uncover regional-specific issues.

3. **Temporal Analysis:** Adding a temporal component to analyze how sentiments evolve over time could help identify patterns related to specific events or seasons.

4. **Handling Neutral Sentiments:** Developing more sophisticated techniques to handle and differentiate neutral sentiments could provide a clearer picture of customer feedback.

5. **Real-time Sentiment Analysis:** Implementing a real-time sentiment analysis system could help airlines respond more promptly to customer feedback and improve service quality dynamically.

6. **Updated geospatial data analysis:** Add a new dataset updated and analyze where the sentiments changed and why.

In summary, the study lays a solid foundation for sentiment analysis in the airline industry, demonstrating the potential of machine learning models in deriving actionable insights from social media data. Further advancements in this field can significantly enhance customer experience and operational efficiency for airlines.

References

1. Aggarwal, C. C., Zhai, C.: A survey of text classification algorithms. Mining Text Data, pp. 163–222 (2012) doi: 10.1007/978-1-4614-3223-4_6
2. Breiman, L.: Random forests. Machine Learning, vol. 45, no. 1, pp. 5–32 (2001) doi: 10.1023/a:1010933404324
3. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27 (1967) doi: 10.1109/tit.1967.1053964
4. Cox, D. R.: The regression analysis of binary sequences. Journal of the Royal Statistical Society Series B: Statistical Methodology, vol. 20, no. 2, pp. 215–232 (1958) doi: 10.1111/j.2517-6161.1958.tb00292.x
5. Friedman, J. H.: Greedy function approximation: a gradient boosting machine. The Annals of Statistics, vol. 29, no. 5 (2001) doi: 10.1214/aos/1013203451
6. Gauch, J. M., Gauch, S., Bouix, S., Zhu, X.: Real time video scene detection and classification. Information Processing and Management, vol. 35, no. 3, pp. 381–400 (1999) doi: 10.1016/s0306-4573(98)00067-3
7. Jurafsky, D., Martin, J. H.: Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition with language models. Pearson (2024)
8. Kaggle: Twitter US Airline Sentiment (2019) www.kaggle.com/datasets/crowdflower/twitter-airline-sentiment
9. Manning, C. D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Cambridge University Press (2008) doi: 10.1017/cbo9780511809071
10. Provost, F. J., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 445–453 (1998)
11. Ramos, J.: Using TF-IDF to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning, pp. 1–4 (2003)
12. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management, vol. 24, no. 5, pp. 513–523 (1988) doi: 10.1016/0306-4573(88)90021-0

Electronic edition
Available online: http://www.rcs.cic.ipn.mx

http://rcs.cic.ipn.mx

Centro de Investigación en Computación