Deep Learning for Classification of 12-lead ECG Signals

Sofia Pinheiro Klautau[1]

[1] Biomedical Engineering Institute (IEB) – Federal University of Santa Catarina (UFSC)
Florianópolis, SC – Brazil

sofia.klautau@posgrad.ufsc.br

Abstract. *This work describes the final project report for the EEL410250 (Machine Learning) course, which involves the classification of cardiac pathologies and normal electrocardiogram (ECG) signals from the PTB-XL database using Convolutional Neural Networks (CNNs). Three trained networks are discussed here: one performing multi-class classification so as to be compared to a base article, and two performing multi-label classification, with one having optimized hyperparameter tuning.*

1. Introduction

Cardiovascular diseases are known to be the leading causes of death in the world (GAIDAI; CAO; LOGINOV, 2023). Early and accurate detection of arrhythmia types, for instance, is extremely important in diagnosing heart diseases and determining a course of treatment (JAMBUKIA; DABHI; PRAJAPATI, 2015). In this sense, artificial neural networks (ANNs) have gained widespread acclaim in classification and detection tasks using ECG signals for a myriad of cardiac pathologies and diagnoses, due to the promising results and possibility of being employed as robust diagnostic aid tools for clinicians (MINCHOLé et al., 2019).

This work's original aim was to use the work done by (ŚMIGIEL; PAŁCZYŃSKI; LEDZIŃSKI, 2021), where Deep Learning (DL) techniques were applied to three different neural network architectures for multi-class classification of ECGs on the PTB-XL database, while training one of them with additional entropy-based features, as a basis for the final project, reproducing the results acquired for the classification of cardiac pathologies using the ECGs. Then, it evolved into also performing multi-label classification, seeing as the database contains multi-label, not multi-class ECG signals. Furthermore, to the best of this work's author's knowledge the code for the original work has not been published.

2. Related Works

Deep Neural Networks (DNNs) have been used for classification of arrhythmia from ECG signals in a level equivalent to that of real cardiologists, as proven by (HANNUN et al., 2019). In the article published by (JAIN; MENON, 2023), windows of ECGs were converted into spectrograms, from which embeddings were generated and used to train different ML classifiers. Their methodology was evaluated using 3 datasets, one of them being the PTB-XL for testing, and they focused on distinguishing abnormal from normal cardiac sinus rhythms.

The study by (AL-ZAITI et al., 2023) presents a novel cohort research using ML models for diagnosing occlusion myocardial infarction (OMI) from ECGs, targeting patients without ST-elevation, who are often missed in initial triage due to a lack of accurate tools to identify them. They provide an OMI risk score, which helped medical workers correctly reclassify one out of three patients with chest pain. Also, the use of pre-trained DNNs and Transfer Learning for ECG classification and detection tasks using signal images has been reported in (RAHHAL et al., 2018). In their work, they describe the application of the continuous wavelet transform to ECG signals in order to generate time–frequency representations to use as the neural network input.

3. Database

The Physikalisch-Technische Bundesanstalt XL ECG database, or PTB-XL (WAGNER et al., 2020), contains 21799 10-second 12-lead ECG signals from 18869 different patients in the WaveForm DataBase (WFDB) format organized on PhysioNet (GOLDBERGER et al., 2000), a community resource for databases of physiological signals. It is one of the largest ECG databases freely available to the public. The signals are multi-label, being comprised of 5 general “superclasses”, as shown in Table 1.

Table 1. Distribution of diagnoses into superclasses.

Records	Superclass	Description
9514	NORM	Normal ECG
5469	MI	Myocardial Infarction
5235	STTC	ST/T Change
4898	CD	Conduction Disturbance
2649	HYP	Hypertrophy

The NORM superclass contains only normal ECGs, while CD is divided into 8 subclasses of specific signals that show conduction disturbances. HYP has 5 subclasses for different types of ventricular, atrial and septal heart hypertrophy. MI contains 4 subclasses for locations of myocardial infarctions, and STTC refers to a change in the ST/T segment in the normal ECG wave, exemplified in Figure 1. A pie chart of the distribution of these super and subclasses is shown in Figure 2.

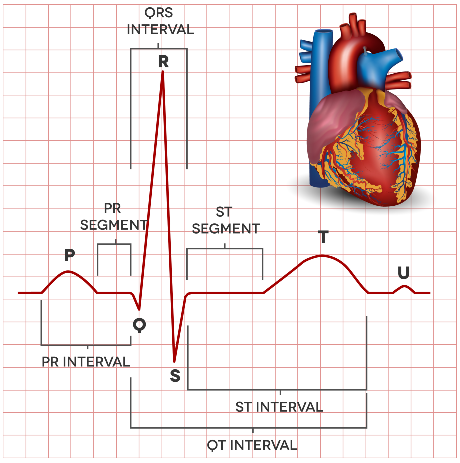

Figure 1. Average normal ECG signal wave. Source: iStock.

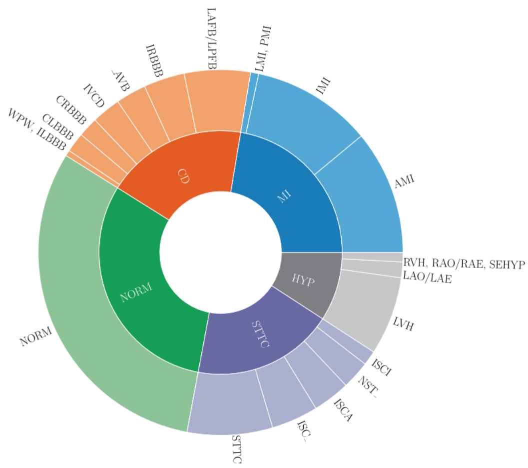

Figure 2. Pie chart of the diagnostic superclasses and subclasses. Source: (WAGNER et al., 2020).

4. Original Work

This section will discuss the original paper (ŚMIGIEL; PAŁCZYŃSKI; LEDZIŃSKI, 2021) used as a basis for this work.

4.1. Data pre-processing

The PTB-XL dataset was pre-processed by filtering out records not classified into diagnostic classes or with less than 100% classification probability. The dataset was then downsampled to 100 Hz and split into training, validation, and test sets in a 70:15:15 ratio. The PTB-XL database now has a new version in which the 500 Hz data is downsampled to 100 Hz. The networks were trained using the Adam optimizer using mini-batches of 128 examples. The learning rate was set at 0.001 at the beginning of the training and was later adjusted to 0.0001. Also, early stopping was employed.

4.2. Classification Task

Three neural network architectures were developed and evaluated in the original work for classification of the ECGs:

1. Convolutional Network: this network consisted of five one-dimensional convolutional layers with LeakyReLU activation functions and a fully connected layer with a softmax activation function, designed to balance computational efficiency with classification accuracy.
2. SincNet-based Network: this network used SincNet layers for low-level feature extraction, followed by two convolutional layers, three fully connected layers, and a final softmax layer. SincNet layers adapt wavelets to raw signal data.
3. Convolutional Network with entropy features: it extended the basic convolutional network by incorporating entropy-based features calculated from each ECG channel. Various entropy measures, including Shannon, approximate, sample, permutation, spectral, SVD and extropy, were used to capture different aspects of the signal's information content.

4.3. Metrics evaluated and results

The metrics used were specificity, defined in Equation 1, sensitivity, defined in Equation 2, F1-score, defined in Equation 4, and accuracy, defined in Equation 5. Also, the area under the Receiver Operating Characteristic (ROC) curve (AUC-ROC) was used.

$$\text{Specificity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}} \tag{1}$$

$$\text{Sensitivity (Recall)} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{2}$$

The AUC-ROC represents the area under the ROC curve plotted with the true positive rate (sensitivity) on the y-axis and the false positive rate (1 - specificity) on the x-axis.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{3}$$

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{5}$$

Tables 2 and 3 represent the results obtained by the authors with the convolutional network and the convolutional network with entropy-based features. Two classes are healthy and sick, the 5 classes are the superclasses aforementioned in Table 1, and the 20 classes are 20 subclasses derived from the superclasses.

Table 2. The results of the convolutional network. Source: (ŚMIGIEL; PAŁCZYŃSKI; LEDZIŃSKI, 2021)

Number of Classes	Acc	Avg AUC	Avg Precision	Avg Recall	Avg F1	Total Params
2	0.882	0.953	0.879	0.882	0.880	8882
5	**0.720**	**0.877**	**0.636**	**0.602**	**0.611**	**11,957**
20	0.589	0.856	0.259	0.228	0.238	27,332

Table 3. The results of the convolutional network with entropy-based features. Source: (ŚMIGIEL; PAŁCZYŃSKI; LEDZIŃSKI, 2021)

Number of Classes	Acc	Avg AUC	Avg Precision	Avg Recall	Avg F1	Total Params
2	0.892	0.960	0.889	0.893	0.891	58,178
5	0.765	0.910	0.714	0.662	0.680	58,259
20	0.698	0.815	0.355	0.339	0.332	58,664

5. Materials and methods

This section will describe the development of the work done for the final project. In order to do this, the convolutional network number 1 in Subsection 4.2 was chosen for reproduction, with the classification of 5 superclasses shown in Table 1 as a target.

5.1. Data Pre-processing

The PTB-XL is present for free on the Kaggle platform in a reformatted version containing comma-separated values (CSV) files and Pickle files, two for each set (training, validation and testing): one with the actual discrete signal values, and another with the metadata (diagnostics and personal information).

They were, in this sense, not divided into a 70:15:15 ratio like in the original work, but followed the recommendation of the database authors of using the column that has a fold number for each signal, which is included in the DataFrame, to do the division. Folds 9 and 10 are considered to have a higher quality due to human evaluations on top of the diagnostic. Also, all records of a same patient are assigned to the same fold, which guarantees that there will be no data leakage. A random division can cause a patient's data to be divided across different sets.

Hence, folds 1-8 were used for training, fold 9 was used for validation and fold 10 was used for testing, exactly like it was suggested. The pre-processing that generates this reformatted database is done by (KHYEH, 2021). It was useful to use the data in this format because the WFDB format would be very time consuming to process.

Then, ECGs that were not classified into any of the 5 diagnostic superclasses were removed from the dataset. Additionally, signals were filtered out if they had less than 20 instances in a certain subclass. The original work did this with a minimum of 100 instances, but it was observed that this higher number would end up removing too many signals of the smaller subclasses.

The signals were normalized using the standard scaler method, with the mean and variance being calculated only using the data from the training set. The ECGs were then converted into arrays so that the array could be integrated into the DataFrame (DF), instead of having multiple lines of the DF contain the different ECG signal points of a same patient. Furthermore, columns containing metadata that would not be used were dropped: age, sex, height, weight, nurse, site, etc. Only the signal itself was used as input data.

From each of the superclass columns, the values were extracted in order to create the label (target) arrays. Finally, since each signal was sampled at 100Hz for 10 seconds, they each have 1000 samples and 12 channels. For this reason, the shape of each one of the label arrays (one for training, one for validation and one for testing) was processed so as to be equal to (x, 1000, 12), where x is the number of signals in each set. The final train set had 17111 signals, with the validation and test sets having 2131 and 2137 signals, respectively.

5.2. Label Encoding

It was noted that while the original authors performed multi-class classification using the 5 superclasses, the ECG signals in the PTB-XL database are in fact multi-label, meaning that they can belong to more than one superclass. Two hypotheses were tested regarding the encoding of the labels as multi-class: firstly, one-hot encoding was performed with the argmax function in order to select only the first positive label from the array as the class label. This was the way found to make the most sense in order to train the multi-class model, and was applied accordingly. Secondly, one-hot encoding was performed without argmax or any other transformation. This led to having a 5x5 array of one-hot encoded arrays for each signal, which is not considered suitable for multi-class classification. In addition, a third hypothesis was created, and the multi-class classification was attempted without encoding of the labels.

For the sake of testing this, a model was defined using the original multi-labeled data and a softmax activation function, so as to attempt multi-class classification without any processing of the data. As expected, the model found it difficult to classify the signals, seeing as it expected one-hot encoded vectors where only one entry is equal to 1, indicating the true class, and all other entries are 0. It suffered from underfitting, with an accuracy below 50%, and the loss function was not minimized at all. Since this test was only done to conclude that the labels had to have been encoded in the original work, its results are not relevant to the continuation of the task at hand and will not be considered for the remainder of this report.

Therefore, the lack of exclusive mutuality between the labels makes it impossible to perform multi-class classification without excluding some of the positive class labels, unavoidably maintaining only one of them even if originally the signal had multiple labels, that is to say, excluding classes from a signal if there are more than one present. The authors of the original work did not disclose how they dealt with this encoding, and did not respond when contacted.

Thus, a contribution of this work is performing also the multi-label classification, considering that each signal can be contained in up to four of the five superclasses. In this way, for said model, the original labels were used without one-hot encoding, seeing as it not necessary to transform them.

5.3. Networks Developed

Initially, the model was trained with a 1-dimensional convolutional neural network architecture as similar as possible to the one described by the original paper, with 5 convolutional layers, 128 as a batch size and a LeakyReLU activation function, which is the ReLu (rectified linear unit function) but with a value for a negative slope instead of a flat one. After training the same network for the multi-class case, in which the softmax activation function in the output layer was used, and for the multi-label case, in which the sigmoid activation function was used, the Optuna framework for hyperparameter optimization was applied to try to find a more efficient network specifically for the multi-label problem, which is considered by this author as the appropriate approach when using the PTB-XL for ECG classification.

In total, various models were trained but only 3 will be compared:

1. Convolutional network for multi-class classification (softmax activation).
2. Convolutional network for multi-label classification (sigmoid activation).
3. Convolutional network for multi-label classification (sigmoid activation) and hyperparameter tuning with Optuna.

6. Results and Discussion

6.1. Multi-class classification (softmax activation)

Using the multi-class model with the labels as described in Section 5.2, the results obtained were of 72.48% test accuracy and an AUC-ROC of 88.46%. This classification was only done to compare the results to those of the paper, which are shown in Table 4.

Thus, this work was able to successfully reproduce the results for the multi-class classification of the 5 ECG superclasses: normal, change in the ST/T segment, myocardial infarction, conduction disturbance and hypertrophy, achieving a test accuracy and AUC-ROC almost equal to those of the original paper. Precision was higher, while recall and F1-score were lower when compared. The training and validation curves can be seen in Figure 3, and the confusion matrix is shown in Figure 4.

Table 4. Comparison of the original paper's multi-class classification and the one done in this work.

Work	Acc	Avg AUC	Avg Precision	Avg Recall	Avg F1	Total Params
Śmigiel et al.	0.720	0.877	0.636	0.602	0.611	11,957
This author	0.7248	0.8846	0.6643	0.5225	0.5404	13,765

Figure 3. Accuracy and loss metrics during training and validation of the multi-class classification.

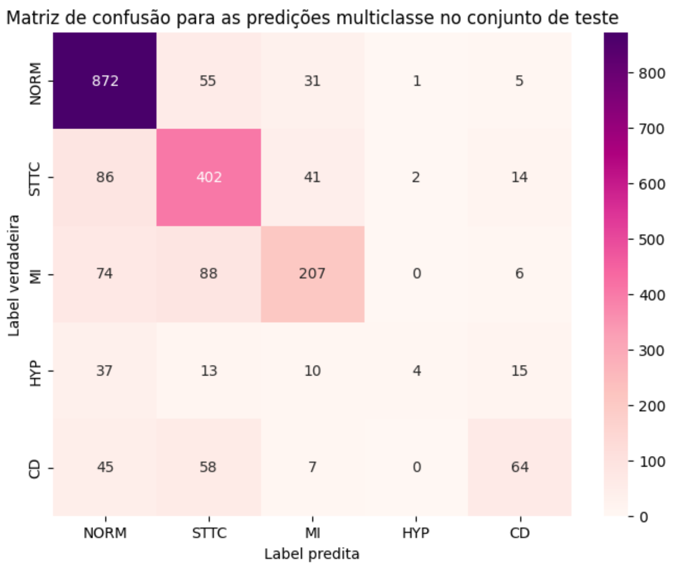

Figure 4. Confusion Matrix for the multi-class classification model.

It's possible to see that there are many misclassifications, namely false negatives in the predicted NORM category, which means that many of the signals are pathological but interpreted as being normal. Meanwhile, the HYP class barely has any positives identified, and the STTC and MI classes suffer from signals being misclassified as false positives. One of the reasons for the misclassifications of these classes is that they may perhaps be very similar and have feature overlap, and may both be present in the same ECG, but with the label encoding only one of them is present. The class imbalance may also play a factor.

6.2. Multi-label classification (sigmoid activation)

A challenge encountered throughout the development of the multi-label classification model was that when the model is trained in the exact same way as the multi-class model, changing only the last activation function, the accuracy computed by Keras is not correct when compared to the accuracy calculated "manually" from the predictions shown in the confusion matrix.

In this sense, it was necessary to implement an accuracy metric that considered that each label must match the corresponding prediction in its exact position for it to be considered correct, and replace the Keras metric. The disparities between the accuracy calculated automatically by Keras and the accuracy "manually" calculated are shown in the code with the latter being called "own accuracy". In this report, only the correct accuracy ("manual") implemented will be considered for the multi-label models.

The architecture for this multi-label model was the same as the multi-class one described in Subsection 6.1, except for the last layer activation function, which for a multi-label classification needs to be the sigmoid so as to independently predict the probability of each class, allowing multiple classes to be assigned to a single instance. The training and validation curves are shown in Figure 5.

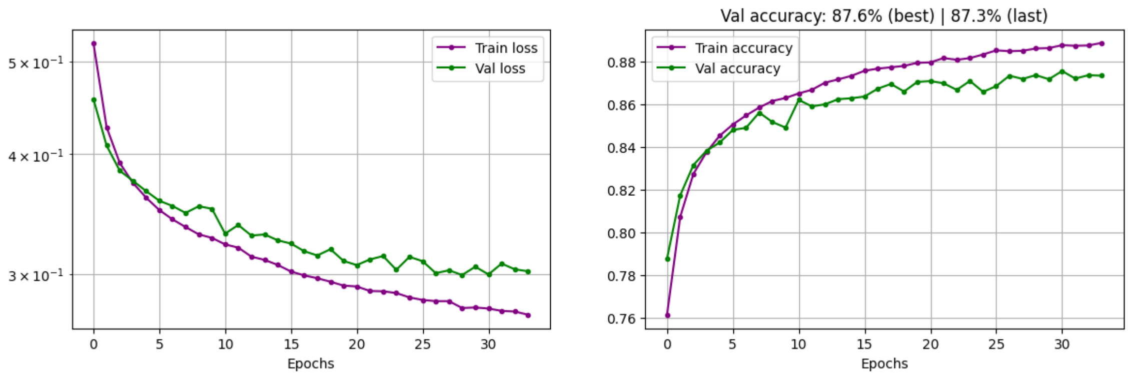

Figure 5. Accuracy and loss metrics during training and validation of the multi-label classification.

The model's results are better than those of multi-class model, with a test accuracy of 87.37% and an average AUC of 90.59%. This behavior might be explained by the fact that the multi-label architecture allows the model to recognize and assign multiple labels to an instance, providing a more nuanced understanding of the data, which is helpful if there is overlap of the features. In contrast, multi-class classification needs to force a single label per instance, which might oversimplify the problem and ignore the coexistence of multiple relevant classes.

The confusion matrices are shown in Figure 6. They are binary, meaning that each one of them shows whether or not the class was found to be present in a given signal. It's noticeable that the true negatives are more correctly identified than the true positives, which means that the model is relatively good at identifying when a class is not present. The classes NORM and MI show higher numbers of true positives, while HYP shows a lower number, which may indicate that it might be a more challenging class for the model to identify.

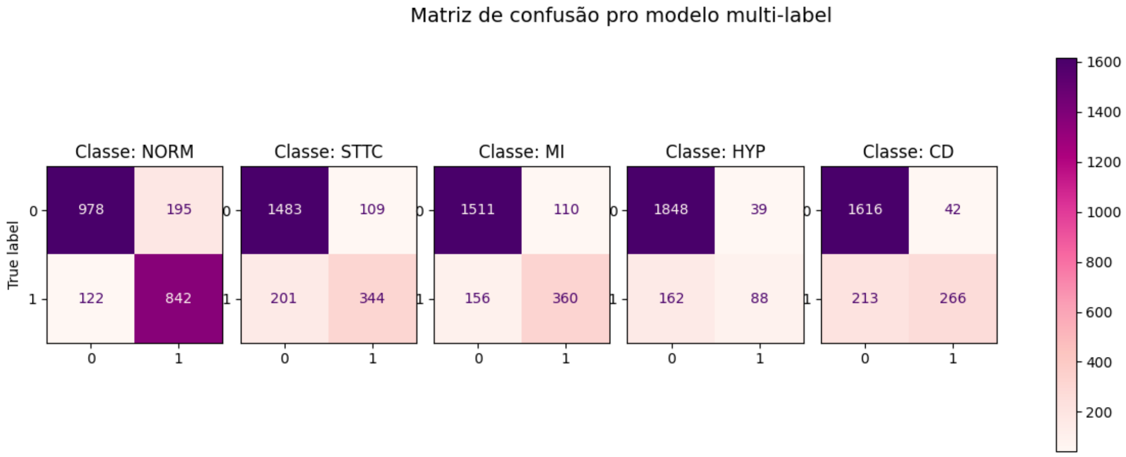

Figure 6. Confusion matrix for the multi-label classification model.

For each class, a ROC curve was plotted and the area under it was calculated, as shown in Figure 7. The AUC values, ranging from 0.89 to 0.93, suggest that the model can discriminate the classes relatively well, with stronger performance for the NORM and STTC classes. The HYP class has the lowest AUC, as expected from the amount of false negatives on the confusion matrix, indicating that this class is harder to classify.

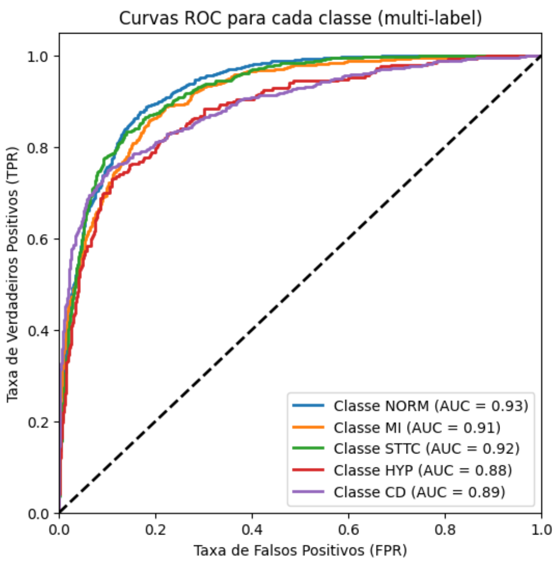

Figure 7. ROC curves and their respective AUC values for the multi-label classification.

In order to have a better interpretation of the results in the healthcare domain, a deeper knowledge of cardiology and general medicine would be imperative. In this way, this project does not delve into the biological conclusions that could be made, since the medical aspect is outside the scope of this work.

6.3. Multi-label classification (sigmoid activation) and hyperparameter tuning with Optuna

Optuna (AKIBA et al., 2019) is a framework for hyperparameter optimization that attempts different hyperparameter combinations set by the user while pruning, or stopping, a certain model that does not seem promising compared to previous results.

In order to attempt to attain higher accuracy and AUC-ROC scores, Optuna was employed to test different model architectures based on the multi-label one described in Subsection 6.2. Some of the hyperparameters optimized were: filter size, kernel size, dropout layer probability, learning rate and activation function. The ranges fed to Optuna for testing are described in Table 5.

Table 5. Range of hyperparameters tested with Optuna.

Hyperparameter	**Range**
Number of layers	4 to 8
Filters per layer	16 to 128 (step: 16)
Kernel sizes per layer	[2, 3, 5]
Activation function	[ReLU, LeakyReLU]
Slope coef. for LeakyReLU	0.008 to 0.15
Dropout rate	0.0 to 0.7
Learning rate	1×10^{-4} to 1×10^{-2}
Batch size	[16, 32, 64, 128, 256]

The optimized network configuration included 6 layers with filter sizes of 112, 64, 128, 96, 96, and 112 for each respective layer. The kernel sizes were 3, 3, 2, 5, 3, and 5, respectively. LeakyReLU was employed as the activation function with a negative slope parameter of approximately 0.0088. A dropout rate of around 0.30 was applied, the learning rate was set to 0.0011 and the batch size was 16. It was noted during training that changing the activation function for the layers did not help with increasing accuracy values, but varying filter size did.

Training the network with hyperparameter tuning improved the performance of the test accuracy metric, albeit not by a great amount. It was 88.79%, while the average AUC-ROC was 92.12%. The validation and training curves are shown in Figure 8.

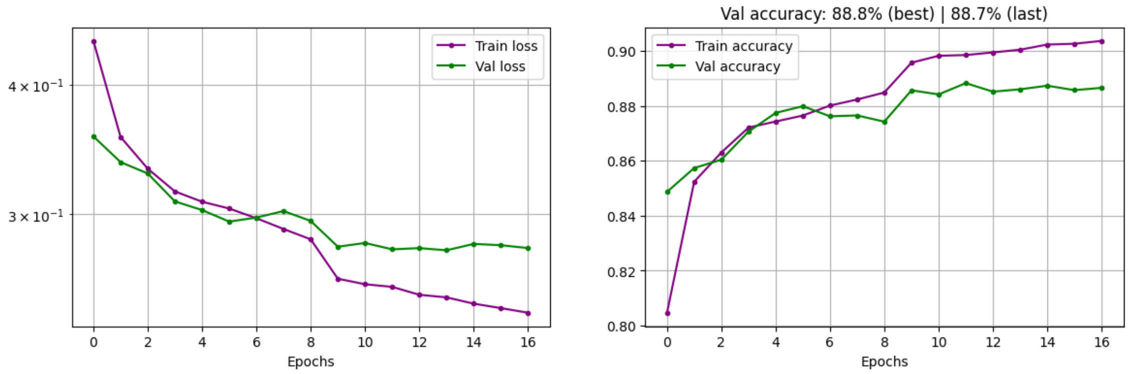

Figure 8. Accuracy and loss metrics during training and validation of the multi-label classification, with Optuna for hyperparameter optimization.

This outcome could be due to the nature of the ECG signals and the fact that they were used raw, without any type of signal pre-processing, such as filtering, that could increase its quality. In future works, it is interesting to pre-process the actual ECG signals before feeding them to the network, as a means to increase performance.

The confusion matrices for the optimized multi-label model can be seen in Figure 9. Its results are similar to those of the first multi-label model, in which the true negative classes are more prevalent in the correct classifications, and the HYP class seems to be more difficult to identify.

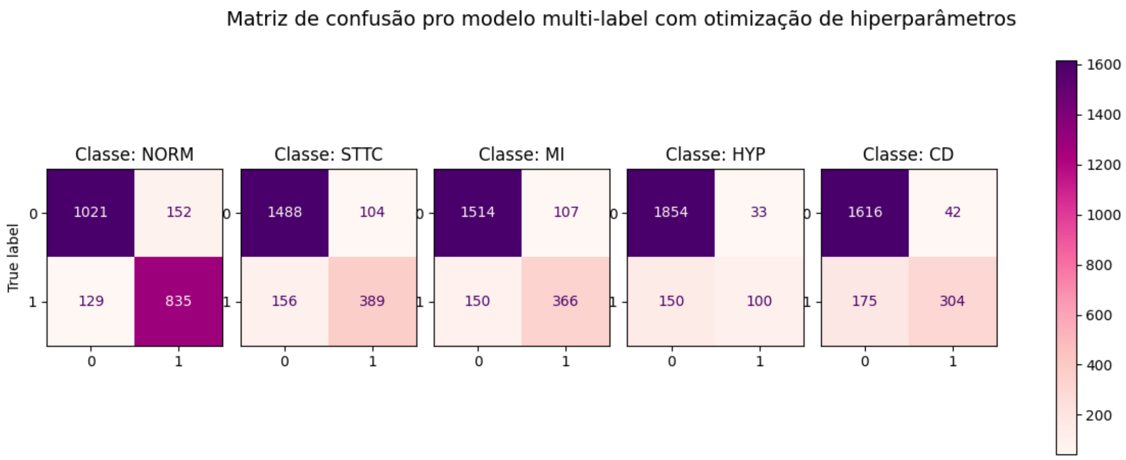

Figure 9. Confusion matrix for the multi-label classification model with Optuna.

The ROC curves and the respective AUC values for each class are shown in Figure 10. Again, the curves confirm that the HYP class seems to be the hardest to be classified.

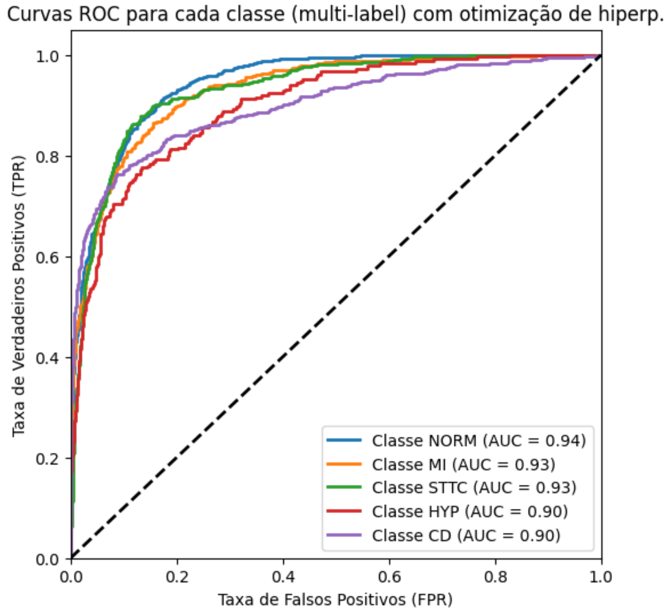

Figure 10. ROC curves and their respective AUC values for the multi-label classification using hyperparameter tuning with Optuna.

Additionally, Optuna's hyperparameter importances plot, shown in Figure 11, highlights which hyperparameters during training most significantly affected the model's performance, in this case, the second and fourth convolutional layer filter sizes, and the dropout rate, respectively.

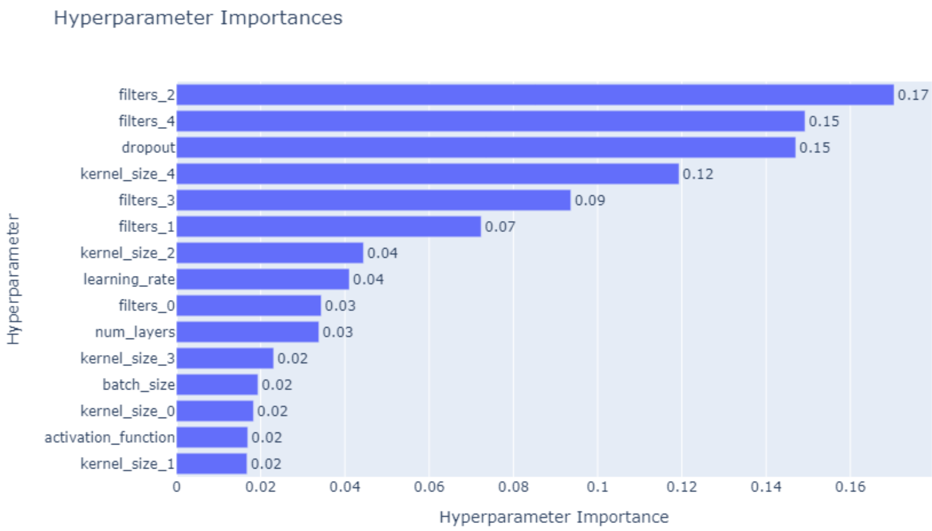

Figure 11. Hyperparameter importances for the optimized multi-label model.

6.4. Summary of the results

Table 6 summarizes the results of the three models compared in this work. The best model was the multi-label with hyperparameter optimization in terms of test accuracy and average AUC-ROC for the curves of each of the 5 classes.

Table 6. The results of the trained networks in this work.

Type	Optimized	Acc	Avg AUC	Avg Precision	Avg Recall	Avg F1
Multi-class	No	0.7248	0.8846	0.6643	0.5225	0.5404
Multi-label	No	0.8737	0.9059	0.7788	0.6219	0.6808
Multi-label	Yes	0.8879	0.9212	0.8079	0.6648	0.7210

7. Conclusion

This work trained three different neural network models, performing 1-dimensional convolutions, to classify ECG signals into five classes from a large, multi-label database. One model was trained for a multi-class problem while two other models were trained for multi-label classification.

Multi-class classification, due to the multi-label nature of the data, is not considered the best methodology, since in order to perform it, the signals must lose some of their classes if there are originally multiple. On that account, the first model was trained in order to try and reproduce the results of the multi-class problem the base work done by (ŚMIGIEL; PAŁCZYŃSKI; LEDZIŃSKI, 2021). The test accuracy and the AUC-ROC for the trained multi-class model were satisfactorily similar to those of the original paper's model.

The multi-label classification, which is considered the most precise approach, was not conducted by the authors of the aforementioned paper, thus it is not possible to compare results with theirs. Nevertheless, the test accuracies and AUC-ROCs achieved by both multi-label models were satisfactory. The multi-label model that was trained with hyperparameter tuning with Optuna obtained the best results, even if they were not much higher compared to the multi-label model trained without optimization.

Overall, this work concludes that before starting any Machine Learning project, it is essential to do a thorough exploratory data analysis on whatever database will used, in order to ascertain that the task in which it will be applied is coherent and will yield factual results. In the biomedical field, it is of utmost importance that cardiac pathologies via ECG, for instance, are correctly detected and classified in exams. If an ML model is not dealing with the signal in a way that considers all the classes of pathologies that it might contain, then it would not be very useful in a real-life clinical decision support system.

A limitation of this work is the lack of proper medical training to be able to interpret the results in a way that allows them to be applicable in the healthcare field. Also, the computation power limits the amount of hyperparameter combinations that can be tested in a timely manner. Furthermore, as previously mentioned, filtering the ECG signals may be invaluable to improve the classification results.

8. Conflict of Interest

The author declares that this final project was not part of another class, dissertation, internship or similar, having been developed exclusively for Professor Danilo Silva's

EEL410250 course at the Federal University of Santa Catarina during the 2024-1 semester.

References

AKIBA, T. et al. Optuna: A next-generation hyperparameter optimization framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. [S.l.]: Association for Computing Machinery, 2019. p. 2623–2631. ISBN 9781450362016. 11

AL-ZAITI, S. S. et al. Machine learning for ecg diagnosis and risk stratification of occlusion myocardial infarction. *Nature Medicine*, Nature Publishing Group US New York, v. 29, n. 7, p. 1804–1813, 2023. 2

GAIDAI, O.; CAO, Y.; LOGINOV, S. Global cardiovascular diseases death rate prediction. *Current Problems in Cardiology*, v. 48, n. 5, p. 101622, 2023. ISSN 0146-2806. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0146280623000397⟩. 1

GOLDBERGER, A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 2000. 2

HANNUN, A. Y. et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, Nature Publishing Group US New York, v. 25, n. 1, p. 65–69, 2019. 1

JAIN, B. U.; MENON, P. G. Classification of abnormal cardiac rhythm from brief single-lead ecg recordings using embeddings from transformer encoder models. In: *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. [S.l.: s.n.], 2023. p. 4899–4901. 1

JAMBUKIA, S. H.; DABHI, V. K.; PRAJAPATI, H. B. Classification of ecg signals using machine learning techniques: A survey. In: *2015 International Conference on Advances in Computer Engineering and Applications*. [S.l.: s.n.], 2015. p. 714–721. 1

KHYEH, K. *PTB-XL Dataset Reformatted*. 2021. Disponível em: ⟨https://www.kaggle.com/datasets/khyeh0719/ptb-xl-dataset-reformatted⟩. 5

MINCHOLé, A. et al. Machine learning in the electrocardiogram. *Journal of Electrocardiology*, v. 57, p. S61–S64, 2019. ISSN 0022-0736. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0022073619304571⟩. 1

RAHHAL, M. M. A. et al. Convolutional neural networks for electrocardiogram classification. *Journal of Medical and Biological Engineering*, Springer, v. 38, p. 1014–1025, 2018. 2

ŚMIGIEL, S.; PAŁCZYŃSKI, K.; LEDZIŃSKI, D. Ecg signal classification using deep learning techniques based on the ptb-xl dataset. *Entropy*, MDPI, v. 23, n. 9, p. 1121, 2021. 1, 3, 5, 14

WAGNER, P. et al. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, Nature Publishing Group, v. 7, n. 1, p. 1–15, 2020. 2, 3

9. Appendix

A bar plot demonstrating the combinations of labels present in the database is shown along the predictions for each new “class” in Figure 12.

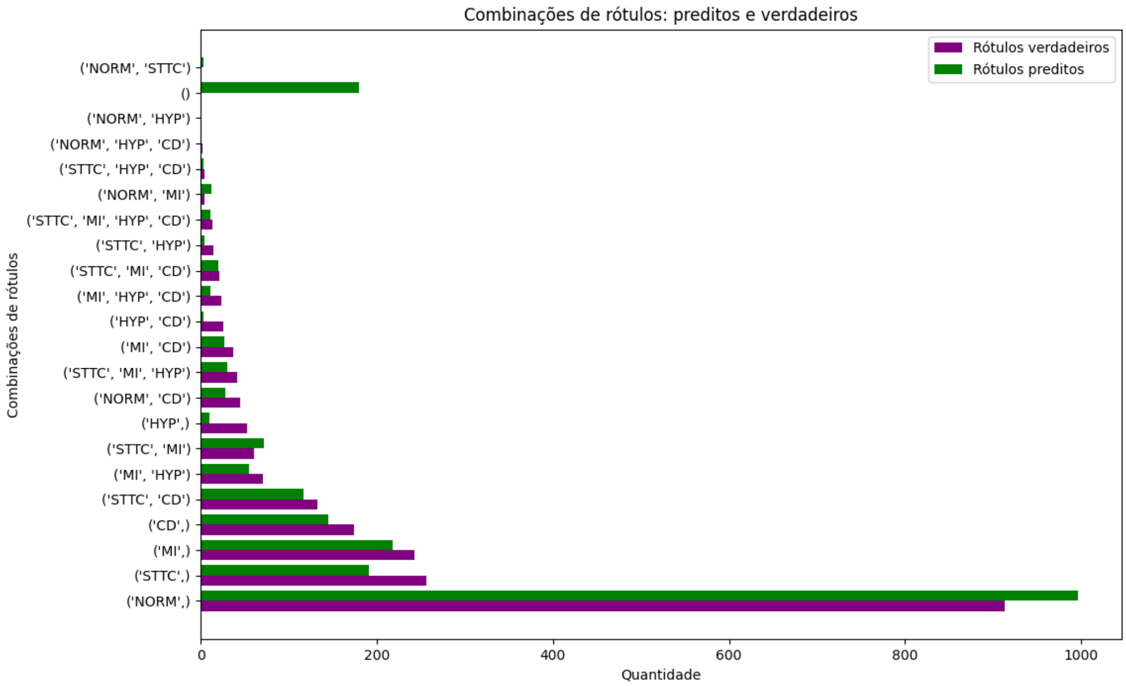

Figure 12. Bar plots comparing the true possible label combinations and the multi-label (not optimized) model predictions.