A Deep Learning Approach for Multiclass Coronary Artery Disease Diagnosis

1st Miguel Silva
Dep. of Mathematics (DMat)
University of Aveiro (UA)
Aveiro, Portugal
mig.silva@ua.pt

2nd Eero Jormalainen
Dep. of Economics, Management, Industrial Eng. and Tourism (DEGEIT)
University of Aveiro (UA)
Aveiro, Portugal
eeroj@ua.pt

***Abstract*—This research work aims to perform a multi-class classification of Coronary Artery Disease (CAD) resorting to the Cleveland Heart Disease Database, distinguishing the various degrees of the disease using Deep Learning. We have developed several models resorting to Deep Neural Networks (DNN) and 1-Dimension Convolutional Neural Networks (1D-CNN). We have used several techniques to boost the performance of the models, such as feature selection for DNN, Synthetic Minority Oversampling Technique (SMOTE), neurons dropout and early stop of training for 1D-CNN. We detected a recurrent problem of overfitting, so after optimization the best model we have obtained was a DNN with 2 hidden layers and 16 neurons per layer (ReLU as activation function for hidden neurons) that underwent SMOTE and dropout of neurons. Precision, accuracy, and recall presented low values, indicating that the classification performance was better when using simpler models, as was done in previous research works.**

***Index Terms*—Coronary Artery Disease, Deep Learning, Multiclass Classification, Deep Neural Networks, 1-Dimension Convolutional Neural Networks**

I. Introduction

Coronary artery disease (CAD) is one of the most deadly heart conditions, according to British Heart Foundation [1]. Therefore, creating Machine Learning (ML) and Deep Learning (DL) models to classify CAD can be crucial, especially in when cases when the medical diagnosis is not found timely. Resorting to the Cleveland Database for Heart Disease [2], some studies have successfully developed ML models to predict the presence or absence of CAD [3] [4], however few have tried to classify CAD according to the degree of severity. Silva & Brito [5] have developed several ML models to perform multiple classification of CAD using Decision Tree (DT), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) algorithms. However, since the classes of the Cleveland dataset were unbalanced they performed several techniques to mitigate the effects of this problem, such as applying the Synthetic Minority Oversampling Technique (SMOTE) to create artificial samples of the classes with lower representation, and determining the class weights with a *Scikit-Learn*'s function [6] and implementing as a hyperparameter of the models. Besides, to reduce the effects of overfitting they have also performed feature selection, where the most effective technique was selecting features randomly through combinations of 3. The Decision Tree algorithms were the one that performed better in all statistical measurements (accuracy, recall, and precision), however they have highlighted that the most important model was the one highest precision due to the nature of the problem. They have achieved a precision performance of 71.91% and the confusion matrix is presented in Figure 1. By analysis of this confusion matrix, they have concluded that even though accuracy is not ideal, the fact that wrong classifications are mostly in the upper part of the confusion matrix is good result due to the nature of the problem.

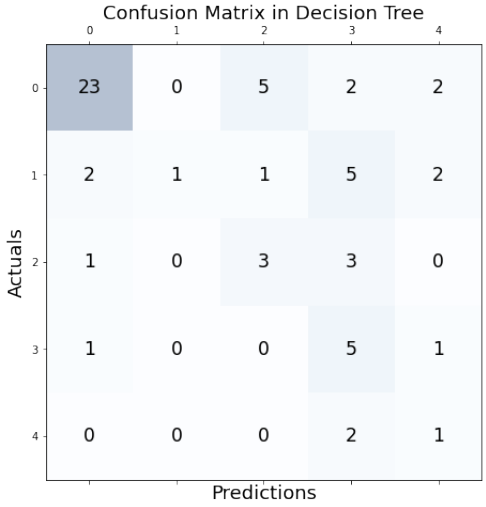


Fig. 1: Confusion matrix of Silva & Brito's best performing model by precision [5].

In this study, we will present an alternative of prediction of the degree of CAD by creating DL models, namely resorting to Convolutional Neural Networks of one dimension (1D-CNN) and Deep Neural Networks (DNN). After tuning the best models and finding the most suitable hyperparameters, we have performed SMOTE on the optimal 1D-CNN model with the best performance, and SMOTE with feature selection to improve the best performing DNN model.

II. Related Work

Miao et Al. [7] investigated the applying deep learning for coronary heart disease (CHD) diagnosis. They proposed a

novel deep neural network (DNN) architecture incorporating a multilayer perceptron with regularization, dropout techniques and a binary Sigmoid classification using deep learning models. The model was trained and evaluated on a dataset of 303 clinical records from patients diagnosed with CAD at the Cleveland Clinic Foundation. The DNN achieved a diagnostic accuracy of 83.67%, demonstrating promising potential for clinical use. Furthermore, the model exhibited high sensitivity (93.51%) for identifying patients with CAD and acceptable specificity (72.86%) for identifying those without. Additional metrics including precision (79.12%), F-score (0.8571), and area under the ROC curve (AUC of 0.8922) support the model's effectiveness.

Shankar et al. [8] proposed a Convolutional Neural Network (CNN) for predicting CAD risk.They used the algorithm to analyze both structured and potentially unstructured patient data. The study acknowledges the impact of incomplete medical data on accuracy, with a decrease expected in such scenarios. The implemented CNN model achieved an accuracy range of 85-88% when evaluated on real-world hospital data. The authors compared the performance of CNN with Nave Bayes, KNN, etc. It is observed that the model accuracy is highest for the model which was designed using CNN with Nave Bayes coming second, and KNN algorithm had the smallest accuracy.

Maluana Azad National Institute of Technology [9] developed a DNN performance this classification task. The authors evaluated its effectiveness against three other classifiers: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Naive Bayes. The model was trained and tested on a the Cleveland Heart Disease dataset. When comparing the different models, SVM learning algorithm achieved the highest accuracy of 86.2%. KNN and Naive Bayes classifiers followed with accuracy of 81.97% and 84.43%, respectively. These findings highlight the importance of algorithm selection in optimizing DNN performance. Additionally, the study explored the influence of optimizer choice and hidden layer configuration. AdaGrad optimizer emerged as the most effective, yielding an accuracy of 85%. Random uniform weight initialization achieved faster training in terms of epochs required. Finally, the investigation revealed a positive correlation between the number of hidden layers and the number of epochs needed for model training. These observations suggest the need for careful consideration of both algorithm selection and hyperparameter tuning to optimize DNN model performance.

Hussain *et al*. [10] have developed a model to classify heart disease based on the same data set as this study, resorting to 1-D CNN to perform binary classification. The architecture of the neural network involved an Embedding Layer to tackle the problem of discrete or categorical variables, two convolutional layers (64 filters) separated by a dropout layer, a global max-pooling layers, and then a fully connected feed forward network to perform classification. The performance of the model was very good, as training and testing accuracy was 97.79% and 96.77%, respectively, precision, recall, and F1-score reached 94.73%, 100%, and 97.29%, and finally the area under the receiver operating characteristic curve was 0.9615. The group has compared the performance of this model with other models created with simpler algorithms, such as logistic regression and random fores, as well as with artificial neural network, and concluded that 1D-CNN was the best algorithm to predict the binary class CAD.

III. Dataset

The Cleveland Heart Disease Dataset [2] has 303 samples from which 6 have missing values, therefore, those samples were excluded. From the 297 patients considered, 96 were female and 201 were male, with age between 29 and 77. As described in Table I, the data set has 13 features (continuous and discrete) and one target variable, which the classification of CAD. The dataset was beforehand divided into a test set and a training set. This division was made by Silva & Brito [5], where they have set aside 20% of samples into a new file, however the division was done making sure the two sets would have the same proportion of classes. The distribution of sets is represented by histograms in Figure 2.

IV. Methodology

A. Deep Learning Algorithms

1) Deep Neural Networks: Deep Neural Networks are network of nodes resembling human neurons and connections between them. They present an input layer where the number of neurons is the same number of features, an output layer where the number of neurons is the same number of classes to predict, and also present hidden layers. These number of hidden layers is what differentiates Artificial Neural Networks from DNN. DNN are deeper, meaning that they have a larger number hidden layers [11]. Each neuron has an activation function, which can be ReLU, Sigmoid, Softmax, and others. The mathematical expressions of the before mentioned activation functions are [12] [13]:

- ReLU: $f(x) = \max(0, x)$
- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$
- Softmax: $\sigma(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \quad for\ i = 1, 2, \ldots, K$

As Neural Networks present sets of weights and biases, the learning process involves optimizing the values of those parameter matrices. Therefore, there are several optimization techniques and the one used in this study is the Adaptive Moment Estimation (Adam). Adam is a method of adaptive learning that stores the exponentially decaying average of past gradients and squared past gradients [14].

Figure 4 presents an example of a standard DNN.

2) 1 Dimension Convolution Neural Networks: Convolutional Neural Networks are traditionally used to learn models from data in image format. In these scenarios and as a general description, several sequential filters are applied by convolution to the image (sample with 3 dimensions), as the image grows in depth and shrinks in height and width, new features are extracted and stored from convolutional layer to convolutional layers. After the sequence of filters, the 3-dimensional image is flattened into a 1-dimension array that

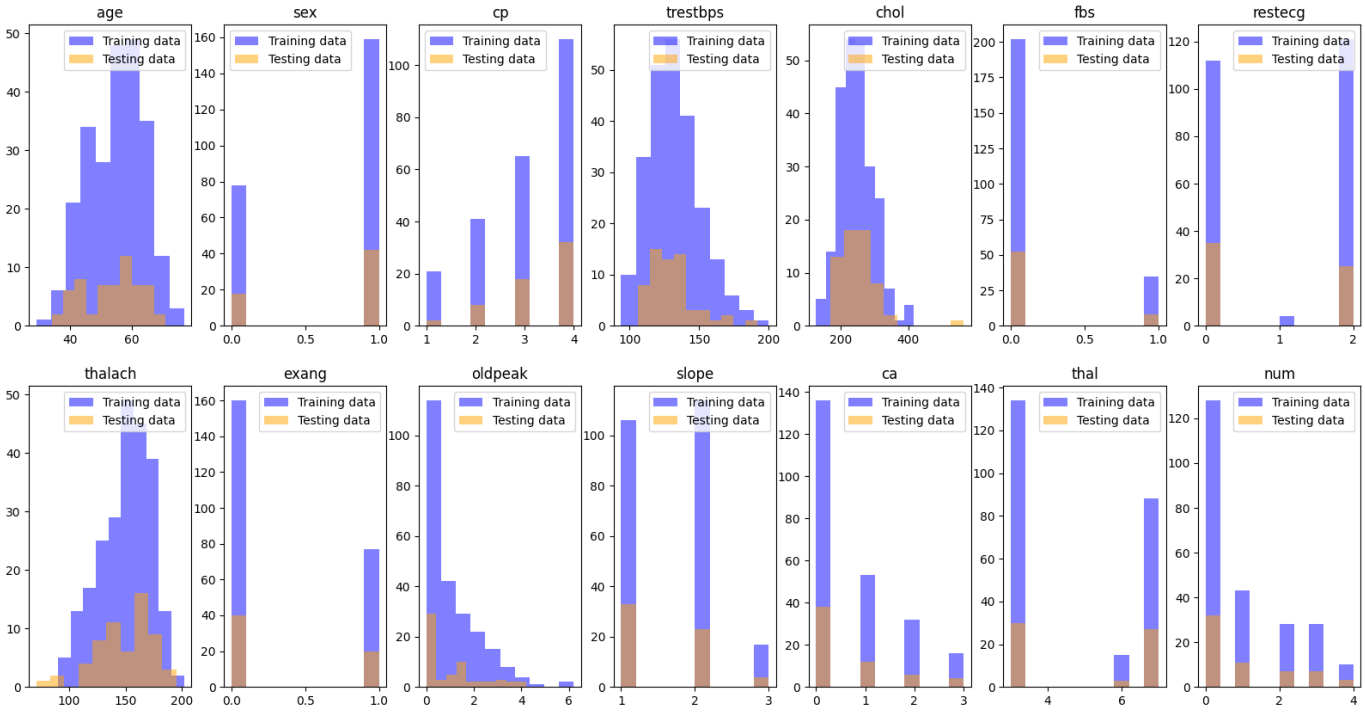


Fig. 2: Distributions of features of training and testing sets.

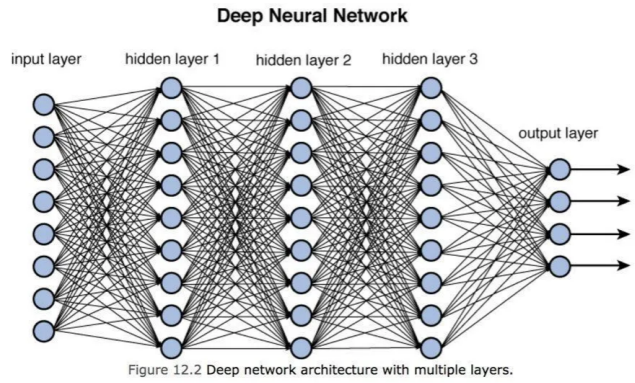


Fig. 3: Example of DNN. [15]

then enters into a fulling connected neural network until it is transformed into an output [16].

In this study, we have used 1-dimensional CNN since our input data was not 3-dimensional. In these types of networks, which are typically used in signal pattern recognition, the input sample and the filters have 1 dimension, growing in depth along the network. After the deepened sample is flattened, it can also be performed pooling where the data can be under-sampled for regularization. In our case, we have performed droupout of 50% of the samples after the convolutional layers. [17]

The activation functions in CNN, both in the convolutional layers and in the fully connected layers, are the ones described before. We have used all the before mentioned functions to fit our models. Also, since the nature of our problem is the classification of a disease, we have opted to compile the model recurring to precision instead of the traditional accuracy, since it is the statistical measure that focuses on evaluating the ratio of true positives, or in other words, how many samples were correctly predicted as positive.

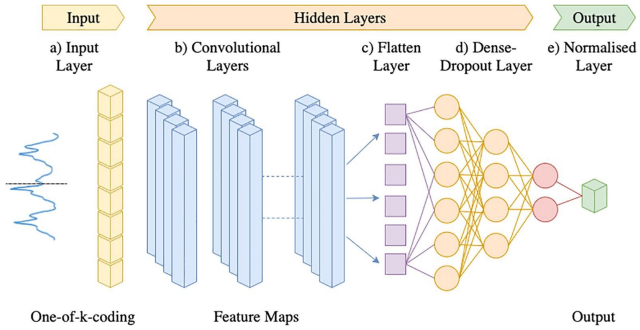


Fig. 4: Example of 1D-CNN. [17]

B. *Feature selection*

Feature selection is a useful method to avoid overfitting the model [18]. Therefore, after performing the model optimization for the DNN models and finding the best hyperparameters, we have performed feature selection to reduce overfitting. Since the variables of this study generally did not present a normal distribution as can be observed in Figure 2, we have ranked the features' importance based on absolute Spearman's correlation by selecting an inter-correlation threshold between pairs of features [19]. We have decided not to perform feature

TABLE I: Description of Features

Feature	Description	Values
age	Age (integer)	
trestbps	Resting blood pressure (mm Hg)	
chol	Serum cholesterol (mg/dl)	
thalach	Maximum heart rate achieved (bpm)	
oldpeak	ST depression induced by exercise	
ca	Number of major vessels colored by fluoroscopy (0-3)	
sex	Gender	0: Female 1: Male
cp	Chest pain type	1: Typical angina 2: Atypical angina 3: Non-anginal pain 4: Asymptomatic
fbs	Fasting blood sugar $>$ 120 mg/dl	0: False 1: True
restecg	Resting electrocardiographic results	0: Normal 1: ST-T wave abnormality 2: Left ventricular hypertrophy
exang	Exercise induced angina	0: No 1: Yes
slope	Slope of peak exercise ST segment	1: Up sloping 2: Flat 3: Down sloping
thal	Thalassemia type	3: Normal 6: Fixed defect 7: Reversible defect
num	Angiographic disease status	0: Absence of CAD 1: First Level of Severity for CAD 2: Second Level of Severity for CAD 3: Third Level of Severity for CAD 4: Fourth Level of Severity for CAD

selection for the 1D-CNN model since the input layer already had a small size (n=13).

C. Class balancing

The dataset we have used in this study presents a significant class imbalance. Therefore, as suggested by Silva & Brito [5], we performed class balancing for the model that presented the best performance. Class balancing was implemented with SMOTE, which is a technique that creates synthetic samples in the vector space corresponding to the misrepresented classes. SMOTE generates synthetic samples in the feature space by computing the distance between a sample and its closest neighbour, and then generating randomly a new synthetic sample that overlaps the line that unites the two original samples. This process is repeated for other samples and nearest neighbors until the misrepresented class has the same number of samples as the prevalent class [20].

D. Evaluation Metrics

As in Silva & Britos's work [5], we have used accuracy, recall, precision, and confusion matrices to test the performance our models and comparing it with their results. It is worth mentioning that since our case of study has multiple classes, recall and precision were calculated through macro averaging [21]. These metrics are computed as follows (TP is the number of true positive predictions, TN true negative predictions, FP false positive predictions, and FN false negative predictions):

- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$
- Recall: $\frac{TN}{TN+FP}$
- Precision: $\frac{TP}{TP+FN}$
- Confusion Matrix: squared matrix that displays the number of TP, TN, FP, and FN for all classes.

E. Dropout and optimization

To avoid overfitting of the models, it is possible to apply dropout of neurons from the network [22]. For DNN, we have used dropout as a tool to boost the model after grid search of hyperparameters. As for 1D-CNN we have set dropout of the last convolutional layers as a standard parameter.

Optimization is the method to adjust the weights and biases of neural networks when fitting the model [14]. Traditionally, Gradient Descent is the optimization function in general linear regression problems and classification models. In this work, we have opted to use Adaptive Momentum Estimation (Adam) because is an effective optimizing function that rectifies vanishing gradients and converges rapidly.

F. Overview

Figure 5 presents an overview of the procedures to optimize the models.

V. Results and Discussion

A. Deep Neural Network

For the DNN-models we applied grid search to find the best parameters and we have fitted the models with precision as the fitting measure, while highlighting the importance of accuracy and recall. Table II shows the parameters of model that achieved the best performance, which were 2 hidden layers with 16 neurons per layer and resorted to ReLU as activation function of the neurons of the hidden layers. The model achieved an accuracy of 0.650, precision of 0.444, and recall of 0.449. In the context of a medical classification task, precision and recall are often used to evaluate the performance of a model in classifying positive cases. The confusion matrix in 6 shows that although the models accuracy did not achieve high numbers, the results in identifying CAD of class 0, 1, and 3 is roughly acceptable, although not good or excellent. However, accuracy can be a misleading performance metric in models with unbalanced datasets as this one. In this can, if the model predicts all test samples as class 0 it can return

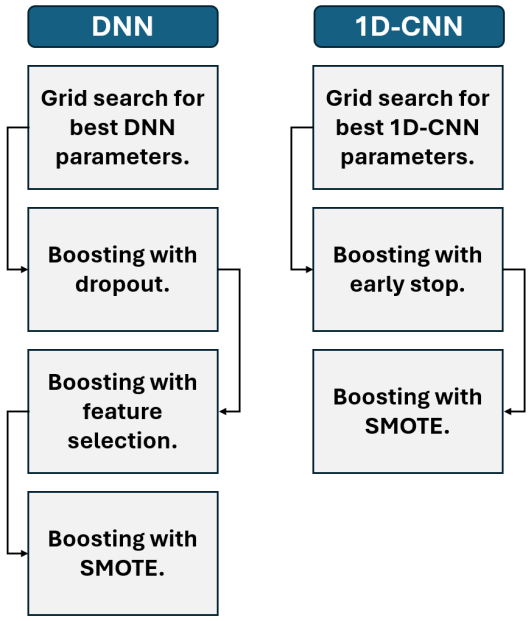


Fig. 5: Overview of the fitting and optmization procedures

an acceptable accuracy value, however it proved that could differentiate any degree of CAD.

TABLE II: Parameter Values and Performance Metrics of best DNN model

Parameters	Value
Number of hidden layers	2
Number of neurons per hidden layer	16
Hidden layers activation function	ReLU
Accuracy	0.650
Precision	0.444
Recall	0.449

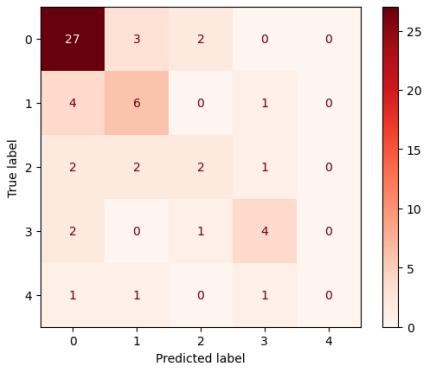


Fig. 6: Confusion matrix of the best DNN model

As we can see from the evolution of the loss function training and testing in Figure 7, the model presented overfitting because it learned fast from training set and performed poor predictions on the validation set [23]. To tackle this problem, we applied a random dropout of neurons on every hidden layers with a factor of $c = 0.3$ to regulate the training. After re-training the model, the problem persisted as seen in Figure 8 with the significant difference in the loss functions of training and validation.

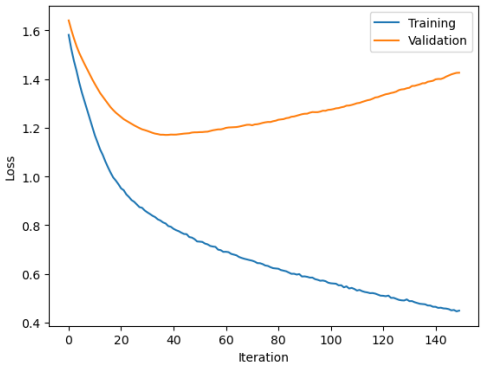
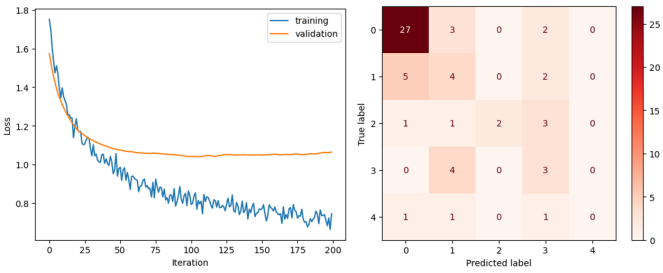


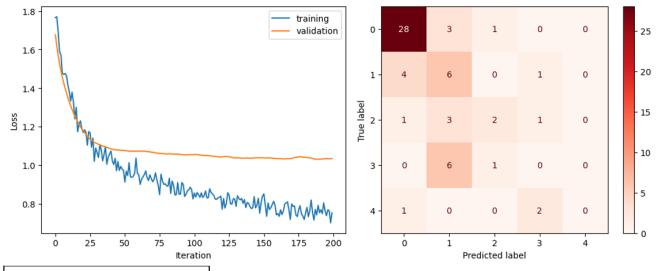
Fig. 7: Loss function of the best DNN model



Accurary:0.600
Precision: 0.475
Recall: 0.384

Fig. 8: Confusion matrix of the best DNN model after dropout (0.3)

To continue to solve the problem of overfitting and maintaining the dropout rate of $c = 0.3$, feature selection through Spearman's correlation was applied, which removed redundant features (highly correlated ones, with a threshold $c = 0.5$), making the model simpler and less prone to overfitting. The results in Figure 9 show that it slightly decreased overfitting, but a loss of precision and recall took place.



Accurary:0.600
Precision: 0.331
Recall: 0.341

Fig. 9: Confusion matrix of the best DNN model after dropout (c=0.3) and Spearman feature selection (c=0.5)

Without solving the problem of overfitting, we have noticed that the loss function of the training set could not get close to 0, therefore SMOTE was applied for the model to be trained with more samples. As mentioned before, SMOTE tries to solve the issue of unbalanced classes by creating synthetic data points for the degrees of CAD with less representation. Combining this method with dropout of neurons from the hidden layers to avoid overfitting potentially created by SMOTE, we have obtained the results presented in Figure 10. Even though accuracy, precision, and recall did not suffer improvements, we observed that the model did not suffer from overfitting and both the training and validation loss functions achieved values closer to 0. The confusion matrix showed us that there is still a considerable number of misclassification, especially considering that the number of samples from class 0 predicted correctly dropped. However, it is visible that the correct predictions from class 1, 2, and 3 increased and the majority of wrong predictions are represented in the upper half of the confusion matrix. This means that when the model wrongly classifies a disease it tends to indicate that the diagnosis is worse that it actually is. Therefore, considering that this is relevant for the clinical context, we assume that is was the best model obtained with DNN.

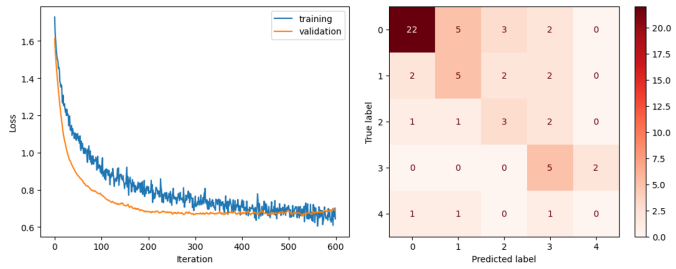


Fig. 10: Confusion matrix of the best DNN model after SMOTE and dropout (0.3)

Silva & Brito have demonstrated that the best model they have developed consisted of three features (*restecg*, *exang*, and *thal*) and the dataset underwent SMOTE. We have also tested a DNN model with these features and SMOTE, however it was observed that the performance did not improve from the results we have presented above.

B. 1-Dimension Convolution Neural Network

For the 1D-CNN models, we have performed grid search to find the optimum values for the number of convolutional layers, the number of filters per convolutional layer, their size, the activation functions of both the convolutional layers and the activation layers of the fully connected network. The results for the best models are presented in Table III and the respective confusion matrix is in Figure 11. Accuracy of 0.633 is a reasonable result in a multi-class problem, as we can observe that most of the cases where CAD was 0 the model predicted correctly. As seen in Figure 12, the model suffered from overfitting, where the validation loss could not approximate to the training loss. However, for scenarios where it was necessary to distinguish from the degree of the disease, the model performed poorly, as most of misclassification were in the lower half of the confusion matrix. This means that the model tends to classify a disease with a lower degree than it actually is.

TABLE III: Parameter Values and Performance Metrics of best 1D-CNN model

Parameters	**Value**
Number of convolutional layers	1
Number of filters per convolutional layer	64
Size of the filters per convolutional layers	4
Convolution activation function	Sigmoid
Fully connected network activation function	Sigmoid
Accuracy	0.633
Precision	0.410
Recall	0.407

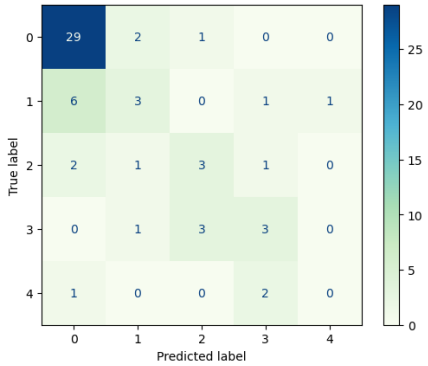


Fig. 11: Confusion matrix of the best 1D-CNN model

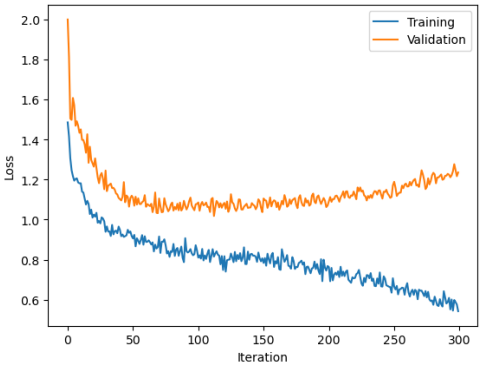


Fig. 12: Loss function of the best 1D-CNN model

In order to improve the model, we have applied several techniques. The first one was early stop, as in problems where overfitting occurs it is advised to stop fitting when validation loss starts to increase while training loss keeps decreasing [24]. Therefore, we have fitted the model with 70 epochs, which is where the distance between the loss functions started to increase, as seen in 12. The loss function and the confusion

matrix after this step are presented in Figure 13. As we can see, performance did not improve, except for a slightly better precision. Besides, the misclassification were still mainly in the lower half of the confusion matrix. This can be misleading in diagnosing CAD.

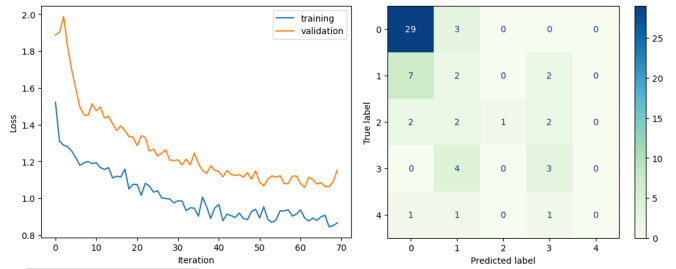


Fig. 13: Confusion matrix of the best 1D-CNN model after early stop (epochs=70)

The second technique we have implemented to improve the model was applying SMOTE to balance the classes and increase the amount of data the model could learn from, as done by Silva & Brito [5]. With SMOTE we have set the number of epochs to n=250 to avoid overfitting. The loss functions and the confusion matrix are presented in Figure 14. As we can see, all performance values decreased with accuracy of 0.533, precision 0.346, and recall 0.347, which was the worst model performance-wise. However, the misclassifications of this model tended to be in the upper part of the confusion matrix, which is relevant for our problem, but this may not be significant as the number of misclassifications was very high. Therefore, the results proved that SMOTE is not a technique that improves the performance in such multiclass classification models, as the model started to overfit before the loss function could approximate to zero. This conclusion confirms what Selvan [25] stated about the fact that SMOTE is a technique that works well mainly with binary classification models.

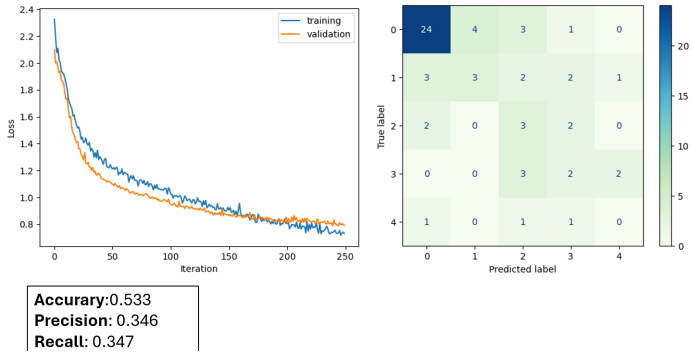


Fig. 14: Confusion matrix of the best 1D-CNN model after SMOTE (epochs=250)

In the end, we observed that it was not possible to improve the work of Silva & Brito [5] through a Deep Learning Approach. This is probably associated with the very low number of samples for the class of higher degree of CAD, as well as the lack of features to build a more complex model with 1D-CNN. It was noticeable in all loss functions that during training, the models could not decrease their error to values close to zero while avoiding overfitting. Therefore, more data is needed for future work.

However, with the DNN model where SMOTE and dropout of neurons were applied, we we have obtained a result similar to the best model's performance created by Silva & Brito. In this DNN model, classes 0, 1, and 3 had good predictions, while classes 2 and 4 suffered to be predicted correctly. Also, the very fact that most of the wrong classifications were false positives (CAD degree higher reality) is a strength of the model, which is clinically more valuable than false negatives (CAD degree lower reality).

VI. Conclusions

Aiming to develop a solution to the classification problem proposed by Silva & Brito [5], we have developed several deep learning models resorting to Deep Neural Networks and 1D Convolutional Neural Networks. The performance of our best models could not present a more robust way of predicting the degree of CAD, as they could only generally distinguish correctly class 0 from the rest of classes. This corroborates the large amount of research work done for binary classification of CAD, and also confirms that the best solution for a potential multi-class classification of CAD may rely on simpler models, such Decision Trees.

VII. Future work

As future work, we propose using a dataset that presents more information, both features and samples, to develop prediction models with higher performance and more accurate capacity of distinguishing various degrees of CAD. Also, Hussain *et al.* [10] used an Embedding Layer to work around categorical layers, therefore we propose evaluating the performance of 1D-CNN models with this method.

References

[1] B. H. Foundation, "Global heart circulatory diseases factsheet," 1 2024.

[2] S. W. P. M. Janosi, Andras and R. Detrano, "Heart Disease." UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C52P4X.

[3] A. B. Nassif, O. Mahdi, Q. Nasir, M. A. Talib, and M. Azzeh, "Machine learning classifications of coronary artery disease," in *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, IEEE, Nov. 2018.

[4] S. Nashif, M. R. Raihan, M. R. Islam, and M. H. Imam, "Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system," *World Journal of Engineering and Technology*, vol. 06, no. 04, p. 854–873, 2018.

[5] M. Silva and J. Brito *Multiclass Classification of Coronary Artery Disease: Machine Learning Approach*, 2023.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] K. H. Miao and J. H. Miao, "Coronary heart disease diagnosis using deep neural networks," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, 2018.

[8] A. Pant, A. Rasool, R. Wadhvani, and A. Jadhav, "Heart disease prediction using image segmentation through the cnn model," in *2023 13th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pp. 374–380, 2023.
[9] V. Sharma, A. Rasool, and G. Hajela, "Prediction of heart disease using dnn," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 554–562, 2020.
[10] S. Hussain, S. K. Nanda, S. Barigidad, S. Akhtar, M. Suaib, and N. K. Ray, "Novel deep learning architecture for predicting heart disease using cnn," in *2021 19th OITS International Conference on Information Technology (OCIT)*, pp. 353–357, 2021.
[11] K. Bharath, "Introduction to deep neural networks," 7 2023.
[12] S. Sharma, "Activation functions in neural networks," *Towards Data Science*, 9 2017.
[13] H. Philips, "A simple introduction to softmax," 5 2023.
[14] S. Doshi, "Various optimization algorithms for training neural network," *Towards Data Science*, 1 2019.
[15] R. Parmar, "Training deep neural networks," *Towards Data Science*, 9 2018.
[16] I. B. M. Corporation, "What are convolutional neural networks?."
[17] R. Sánchez-Reolid, F. López de la Rosa, M. T. López, and A. Fernández-Caballero, "One-dimensional convolutional neural networks for low/high arousal classification from electrodermal activity," *Biomedical Signal Processing and Control*, vol. 71, p. 103203, Jan. 2022.
[18] R. Pramoditha, "How to mitigate overfitting with feature selection," *Towards Data Science*, 10 2021.
[19] A. J. Bishara and J. B. Hittner, "Testing the significance of a correlation with nonnormal data: Comparison of pearson, spearman, transformation, and resampling approaches.," *Psychological Methods*, vol. 17, p. 399–417, Sept. 2012.
[20] C. Maklin, "Synthetic minority over-sampling technique (smote)," 5 2022.
[21] E. AI, "Accuracy, precision, and recall in multi-class classification."
[22] A. S, "Dropout: A powerful regularization technique for deep neural networks," *LinkedIn*, 3 2023.
[23] B. Carremans, "Handling overfitting in deep learning models," *Towards Data Science*, 8 2018.
[24] R. Pramoditha, "Using early stopping to reduce overfitting in neural networks," *Data Science 365*, 7 2022.
[25] T. Selvan, "How to handle multiclass imbalanced data?- say no to smote," *Towards Data Science*, 8 2020.