Final Report Project

Advanced Machine Learning

ECE 5424 Fall22

A Machine Learning Framework Coupled with Data Augmentation Algorithms to Speed up the Diagnosis of Coronary Artery Disease

Professor Creed Jones

Group H:

Elham Nasarian

Vivek Joshi

Sakshi Taori

December 2022

Duties and Responsibilities

Part	Paper	Python Code
Heart Disease Dataset (Providing & Description)	Elham	-
Abstract	All members	-
Introduction	All members	-
Related Works	All members	-
Proposed Framework	All members	-
Autoencoders (AEs)	Elham & Sakshi	Elham
SMOTE	Sakshi	Sakshi
Data Preparation	Sakshi & Vivek	Sakshi & Vivek
One hot encoding	Vivek	Vivek
Thresholding	Sakshi & Vivek	Sakshi & Vivek
Random Forest	Elham	Elham
XGBoost	Elham	Elham
Logistic Regression	Sakshi	Sakshi
ANNs	Vivek	Vivek
Tables & Figures	Elham	All members
Results	All members	All members
Discussion	All members	All members
Conclusion	All members	All members
References based on IEEE	Elham	All members
Revise	All members	All members

A Machine Learning Framework Coupled with Data Augmentation Algorithms to Speed up the Diagnosis of Coronary Artery Disease

ECE5424 Professor Creed Jones

Elham Nasarian
Department of Industrial and System Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA, US
elhamn20@vt.edu

Vivek Joshi
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA, US
vivekj@vt.edu

Sakshi Taori
Department of Electrical and Computer Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA, US
sakshit@vt.edu

***Abstract*— Every 36 seconds, cardiovascular disease kills a person in the United States. Coronary artery disease (CAD) is the highest frequent kind of cardiovascular disease in United States [1]. CAD leads to staggering annual healthcare expenditures of $5.54 billion [2]. Due to the fact that many people's first experience with CAD is a heart attack, early and effective identification of CAD using risk factors that could lead to its advancement is crucial. Angiography is the best and most common way to predict CAD, but it has many side effects and is expensive. Machine Learning Methods can be used in this context to predict CAD, and several models and datasets have been improved for this purpose. However, only few studies have explored the diagnosis of CAD in the individual arteries. In this project, we focused on problem of stenosis in individual LAD, LCX, and RCA by applying Random Forest, XGBOOST, Llogistic regression, and Artificial neural network on the Z-Alizadeh Sani dataset that comprised 303 subjects, each with 54 features. Moreover, our proposed framework is developed to handle data scarcity in the diagnosis of CAD. Herein, we initially performed data augmentation with Autoencoder (AE) and SMOTE, and generated new datasets, further comparing the accuracy rates of these machine learning models on the generated datasets with the original dataset. The experimental results revealed that the average accuracy of the AEs for diagnosis of stenosis in individual RCA, LCX, and LAD, achieving accuracy rates of 91.37%, 94.45% and 94.19%, is higher than that of the SMOTE. To demonstrate the generality of our augmentation methods, we trained AML prediction methods on our dataset (with and without data augmentation) and compared their performances.**

Keywords—Advanced machine learning, data augmentation, coronary artery disease, diagnosis, accuracy.

I. INTRODUCTION

Plaque buildup in the coronary arteries and other arteries throughout the body is the cause of coronary artery disease (CAD). Plaque is composed of cholesterol and other chemicals deposited in the artery. Plaque accumulation narrows the interior of the arteries over time, which can partially or completely obstruct blood flow. Due to the fact that constricted arteries can block blood flow to the heart and the rest of the body, they can cause chest pain. This condition is known as atherosclerosis. For many patients, a heart attack is the first indication that they have CAD. Heart attack symptoms include chest pain or discomfort (angina), weakness, lightheadedness, nausea, or a cold sweat, pain or discomfort in the arms or shoulders, and shortness of breath. CAD can weaken the heart muscle over time. This may result in heart failure, a dangerous condition in which the heart cannot pump blood properly. The well-known risk factors for CAD are overweight, physical inactivity, unhealthy eating, and smoking tobacco. In addition, a family history of heart disease, especially early-onset heart problems (50 or younger), raises the risk for CAD. Several clinical indicators, including blood pressure, blood cholesterol, and blood sugar levels, are commonly tested to identify the risk of coronary artery disease. For screening patients suspected of having CAD, experimental techniques such as angiography, echocardiography, and magnetic resonance imaging (MRI) are often recommended for a more accurate diagnosis. These more accurate procedures, however, may have major side effects and are costly. Allergic responses to the local anesthetic, contrast dye, or sedative, bleeding, bruising, blood clots, injury to an artery or vein, damage to the heart walls, and infection are some of the side effects reported in the literature. The coexistence of side effects and increasing burden of healthcare expenditure have motivated many researchers to apply Machine Learning (SL) and Data Mining (DM)algorithms for quick and accurate detection of CAD.

Heart has three coronary arteries, and we can say someone has CAD, if one of these blood arteries has stoned: the Left Anterior Descending (LAD), Left Circumflex (LCX), and Right Coronary Artery (RCA) [3]. There have few studies for predicting in each artery, for this reason we decided to apply our new framework for each of LAD, LCX, and RCA. For this reason, we have used the ZAlizadehSani coronary artery disease dataset that can be found in UCI Machine Learning Repository [4].

We used Machine Learning models coupled with Data Augmentation algorithms for these three arteries. The ML algorithms are: RF, XGBOOST, LR, and ANN. In our study, we proposed a ML framework with Data Augmentation technique that includes Synthetic Minority Over-sampling technique (SMOTE), and Autoencoder coupled with Thresholding, and One-Hot-Encoding. The methodology includes five steps; first - apply ML methods on original data with 303 rows, second - use autoencoder to augment data

with the ratios (10:1~3000 rows), third - use autoencoder to augment data with (20:1~6000 rows), fourth- use autoencoder to augment data with (25:1~7500 rows), and fifth- use SMOTE algorithm to augment the data with 432 rows. The data augmentation proposed framework can improve the performance of CAD prediction with ML methods. To compare with related works, we have gained the highest results for LAD, LCX, and RCA in the ZAlizadehSani coronary artery disease dataset.

In the following section, we look at the related work in Section II. We discussed about our framework with details on material and methods in Section III. The final results are summarized in Section V, and talked about conclusion and future works in Section VI

II. RELATED WORK

Abdar et al. [5] argued that due to its significant impact on the society, early and accurate detection of CAD is essential. The study proposes a novel model which combines several traditional machine learning methods (decision trees, artificial neural networks) and deep learning approach for effective diagnosis of CAD. The model is validated using two well-known CAD datasets (Z-Alizadeh Sani and Cleveland). To improve the performance of the model, some clinically significant features selected from the datasets using a genetic search algorithm and finally a multi-level filtering technique is applied to balance the data using the ClassBlancer and Resample methods. The findings show that the approach provides the accuracy of 94.66% and 98.60% to predict CAD entities in the Z-Alizadeh Sani and Cleveland CAD datasets, respectively.

Aouabed et al. [6] tackles the problem of CAD detection using a new accurate hybrid machine learning model. The proposed ensemble model combines several classical machine learning techniques including Decision trees, Support vector machines, Artificial neural networks (ANNs). The base algorithm is used with four different kernel functions (linear, polynomial, radial basis and sigmoid). Authors used the model to analyze the Cleveland CAD dataset. To improve the performance of the model, the most important features of the dataset are identified using a genetic search algorithm. Tama et al. [7] emphasized that owing to the fact that a heart attack occurs without any apparent symptoms, an intelligent detection method is fruitful. Authors proposed a novel CHD detection method based on a machine learning technique (i.e., Random Forest, Gradient boosting machine, and Extreme gradient boosting). Next, their detection model is evaluated on multiple heart disease datasets, i.e., Z-Alizadeh Sani, Statlog, Cleveland, and Hungarian. In addition, a particle swarm optimization-based feature selection is carried out to choose the most significant feature set for each dataset.

As an alternative to the available diagnosis tools/methods (e.g., Angiography), Ghiasi et al. [8] introduced a decision tree learning algorithm called classification and regression tree (CART) and the results are compared with other models in the literature (e.g., Sequential Minimal Optimization (SMO), Naïve Bayes (NB), artificial neural network (ANN), Bagging, and genetic algorithm (GA). Authors contended that CART can outperform some of existing models using Z-Alizadeh Sani dataset.

AlizadehSani et al. [9] developed a model to handle model uncertainty in the prediction of individual artery stenosis. Their results demonstrate high diagnostic performance of the proposed method for diagnosis of stenosis in individual RCA, LCX, and LAD, achieving accuracy rates of 82.67%, 83.67% and 86.43%, respectively. In the end, they used the genetic algorithm (GA) to determine the hyper-parameters of the support vector machine (SVM) kernels.

Nasarian et al. [10], introduced a new hybrid feature selection algorithm called heterogeneous hybrid feature selection (2HFS). In this work, they used Nasarian CAD dataset, in which work place and environmental features are also considered, in addition to other clinical features. Synthetic minority over-sampling technique (SMOTE) and Adaptive synthetic (ADASYN) are used to handle the imbalance in the dataset. Decision tree (DT), Gaussian Naive Bayes (GNB), Random Forest (RF), and XGBOOST classifiers are used. They have received accuracy rate of 81.23% with SMOTE and XGBOOST classifier. They have also tested their approach with other well-known CAD datasets: Hungarian dataset, Long-beach-va dataset, and Z-Alizadeh Sani dataset. We have obtained 83.94%, 81.58% and 92.58% for Hungarian dataset, Long-beach-va dataset, and Z-Alizadeh Sani dataset, respectively.

Khozeimeh et al. [11], proposed a model, called the CNN-AE, to predict the survival chance of COVID-19 patients using a CNN trained with clinical information. A data augmentation procedure based on autoencoders (AEs) was used to balance the dataset. The They have gained the accuracy of the CNN-AE (96.05%) was higher than that of the CNN (92.49%). They trained some existing mortality risk prediction methods on their dataset (with and without data augmentation) and compared their performances.

Acharya et. al [12], consisting of ECG signals from 40 normal and 7 CAD patients. Two ECG segmented signals of Net 1 (2 seconds) and Net 2 (5 seconds) were used with a total of 93500 and 38120 samples, respectively in each set. They predicted the presence of CAD in both Net 1 and Net 2 using convolutional neural network (CNN) structures with four convolutional layers, four max pooling layers, and three fully connected layers. This deep learning network achieved an accuracy of 94.95%, sensitivity of 93.72%, and specificity of 95.18% for Net 1 (two seconds). The model for Net 2 dataset resulted in an accuracy of 95.11%, sensitivity of 91.13%, and specificity of 95.88%.

III. PROPOSED METHODOLOGY

In this study, our new framework is developed to handle data scarcity in the prediction of individual artery stenosis. Two well-known data augmentation algorithms, Autoencoders (AEs), and Synthetic Minority Oversampling Technique (SMOTE) and four ML methods were selected to investigate the CAD dataset. We further performed Thresholding operation and One-Hot-Encoding for post-processing on the new augmented datasets.

Moreover, we applied four classification ML algorithms, Random Forest (RF), Logistic Regression (LR), Extreme Gradient Boosting (XGBOOST), Artificial Neural Networks (ANNs). Since, CAD diagnosis is a biomedical application, the focus is higher on reducing the false negatives (increasing sensitivity). Therefore, while evaluating our models, we looked at the confusion matrix and chose the architectures

which had the least number of false negatives. Figure 1 shows our proposed framework.

As Fig. 1 shows, initially two Augmentation techniques (AEs and SMOTE) are used to increase the size of the original dataset. As the dataset had categorical features, one-hot encoding was performed. The dataset features were not balanced; therefore, thresholding was used to map the exact ratio of the original datasets with the augmented datasets. After using these methods, we applied four advanced machine learning models to predict CAD, RF, LR, XGBoost, and ANNs. Then, we compared the results, and selected the best model. In the following parts, we talk about each method briefly.

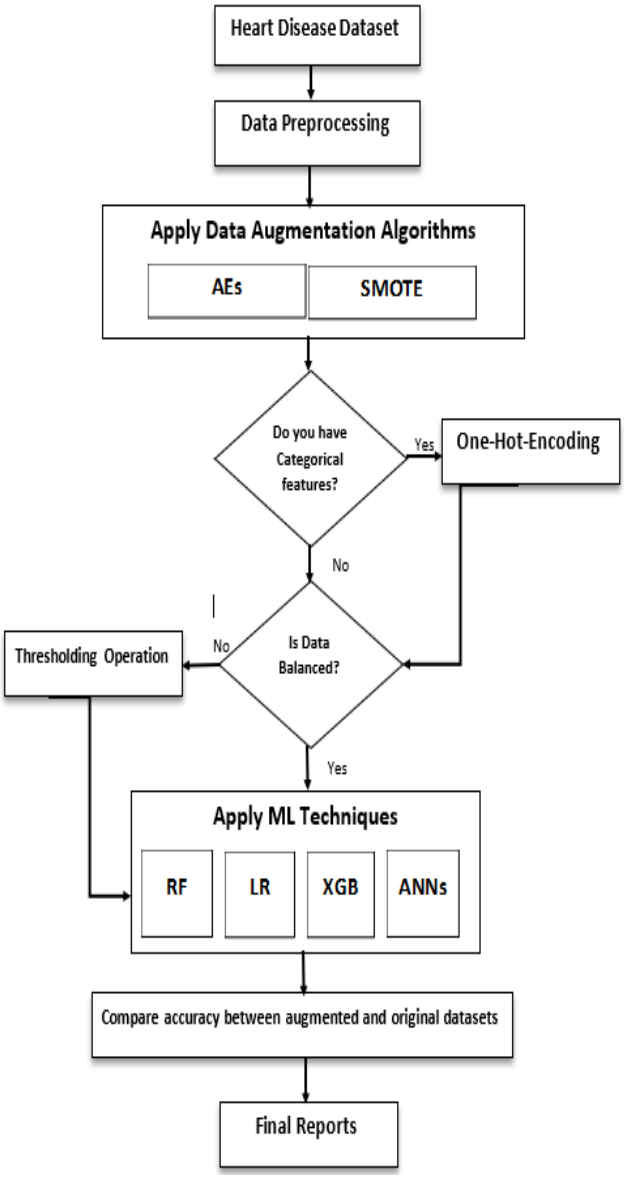

Figure 1. The proposed framework for CAD prediction.

A. Autoencoders (AEs)

To ensure accurate classification, it was necessary to balance and increase the size of dataset. To increase the number of data, first, an AEs model was used. To carry out the data augmentation, the 58 features of the original dataset were fed to the AE to undergo the compression and decompression routines [11].

AEs belong to the realm of unsupervised learning, as they do not need labelled data for their training. In brief, an AE compresses input data to a lower dimensional latent space and then reconstructs the data by decompressing the latent space representation. Similar to principal component analysis (PCA), AEs perform dimensionality reduction in the compression phase. However, unlike PCA, which relies on linear transformation, AEs carry out nonlinear transformation using deep neural networks. Figure 2 shows the architecture of a typical AE [11].

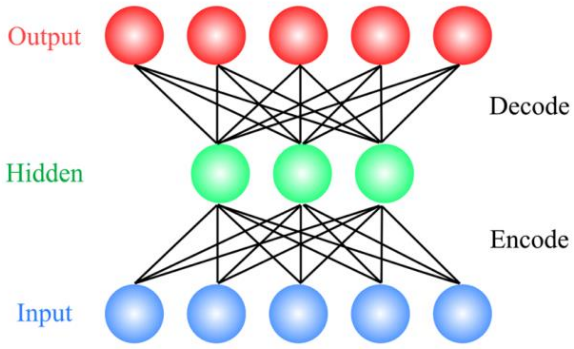

Figure 2. AE architecture [11]

Initially, a function for the Autoencoder model was developed using the TensorFlow Neural Network library. There was a total of three layers with one hidden layer. The sizes of the input and output layers were 58, equal to the entire count of features in the original data. The hidden layers were of size 32, that is lower than 58 since the hidden layer of the autoencoder is smaller than the input and output layers. Adam optimizer was used for training the model and the activation function for all three layers was RELU. The loss function was set as binary cross-entropy. This model function was further provided as an input to another custom augmentation function that was specifically used to generate new augmented data from the given input dataset. In the main code, the augmentation function was implemented inside a for loop. On a single execution of the loop, 303 new rows (equal to the samples in original dataset) were generated. This output augmented data was further concatenated with the original dataset. This process created the final dataset (303+303 = 604 rows). Based on the number of times the loop is executed, datasets with different augmentation ratios can be obtained. Steps used in the augmentation process were:

1. For data pre-processing, the original dataset was imported as a data frame and divided it into two smaller data frames; one containing categorical data and the other with just the binary feature variables. As the categorical features should not be scaled, after imputation, the binary data frame was scaled

using the MinMax scaler and concatenated with the categorical data frame.

2. This initial dataset was provided as an input to our autoencoder function, which generated datasets with different augmentation ratios.
3. The following calculation was performed to get the augmentation data for a specific ratio:
   Augmentation ratio: X:1
   Total number of samples: 303*X
   (303 is the number of samples in initial dataset)

As running for loop once gave 303 new samples that are concatenated with the 303 original samples. To obtain 303*X samples in total, the loop will be run for:

N = (303*X - 303)/303.

The augmentation ratios chosen for this study were (10:1 ~ 3000 samples, 20:1 ~ 6000 samples, and 25:1 ~7500 samples). Based on the value of N for each case, augmented datasets were generated using autoencoder.

B. Synthetic Minority Oversampling Technique (SMOTE)

The presence of an imbalanced dataset is a potential source of bias for any machine-learning model in classification. To overcome this, two types of sampling techniques are used: under-sampling and over-sampling. While under sampling focuses on reducing the majority class samples, oversampling focuses on increasing the minority class samples to make them equal to the majority class. SMOTE is an oversampling technique that was first introduced by [12]. This technique utilizes k-nearest neighbors to generate synthetic samples. Initially, a value N is evaluated based on the number of required oversampling observations to make the majority-to-minority class ratio 1:1. Further, a random sample from the minority class is chosen and its k-nearest neighbors (k=5, default). This is followed by calculating the distance between the random sample and its nearest neighbors. After multiplying this distance metric by any random number between [0, n], a synthetic sample is created. In this way, k-nearest neighbors are used for interpolating N synthetic samples of the minority class.

This technique is primarily used for data balancing. Data augmentation is used for creating artificial data points by applying some transformations to the existing dataset. Hence, SMOTE can also be considered a data augmentation technique because, in the process of balancing the dataset, it generates synthetic samples of the minority class, which results in increasing the overall size of the original dataset. Figure 3 shows the architecture of SMOTE.

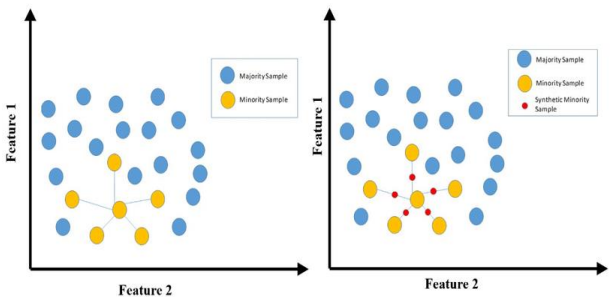

Figure 3. SMOTE architecture [13]

Our project dataset has a binary target variable (CAD) that indicates the presence or absence of coronary artery disease. Further, we aimed to predict the binary variable indicating stenosis in the three individual arteries (LCA, LCX, RCA). As the total number of samples in our dataset is less {303: 216 (CAD), 87 (Normal)}, the data augmentation process was carried out using SMOTE technique. We used the over_sampling.SMOTE library from imblearn with the value of sampling_strategy as 1. 1 indicates that there will be an equal number of samples in the majority and minority class in the final dataset. At first, the Min-Max scaling was performed on the binary feature variables. The binary feature vector was then concatenated with the categorical features and given as input to the SMOTE function. This operation resulted in a final augmented dataset with {432 rows: 216 (CAD), 216 (Normal)}. So, the number of samples increased from 303 to 432.

After implementing this technique, we further explored ways to augment the dataset to a larger number of ratios (10:1, 20:1, 25:1). We realized that a for loop needs to be executed where the original dataset has to be subsampled every time, and SMOTE should be used. This will generate the augmented subsampled dataset, which we can combine with our original dataset to create a larger dataset. This process has to be repeated till the desired number of samples is obtained. This process was time-consuming and hard to code. As we were able to get the augmentation datasets using Autoencoder, the repeat implementation of SMOTE was not performed. However, the initially generated SMOTE augmentation dataset with 432 rows has still been included in our analysis. Its performance will be compared with the original dataset and augmentation datasets generated by the Autoencoder.

C. Data Preparation and One hot encoding:

Since the features in the dataset have different ranges, we need to normalize the data so machine learning models can work properly on it and not get skewed by one feature with large values. To do this, we apply a simple min-max scaler to the dataset to scale all numerical values in the dataset to a range of -1 to +1.

After the normalized data was augmented, there were still a few steps left before we could actually apply various models and compare the results. Firstly, there was a column called ‘Exertional CP’, that had only one value throughout the dataset. A column like this will have no correlation with other columns, and hence, should be dropped, as it would only increase the complexity and training time without actually helping in increasing the accuracy. We also found that we have four categorical columns in our dataset, viz, Functional Class, BBB, Region RWMA and VHD. These columns do not work well with machine learning models in their current state, so we had to use one hot encoding on them. This resulted in creation of multiple new columns, that contained numeric data instead of categorical one’s. The original categorical data containing columns were then dropped. The usage of one hot encoding leads to an increase in the number of dimensions of the dataset, which could lead us to the curse of dimensionality problem. To deal with this problem, feature selection or principal component analysis is usually used, but since even after adding the newer columns, out dataset only

had 74 columns, we did not feel the need to apply feature selection to reduce the size of the dataset. Hence, out models were trained on the entire dataset containing all 74 columns.

D. Thresholding:

The augmented data resulted in continuous values for columns that were supposed to be binary or categorical. Categorical columns have values divided into various categorical classes represented by whole numbers, hence having continuous data present here does not make any sense. For categorical data, the augmented values were just rounded off to the nearest integer to represent the corresponding categorical class. For binary data on the other hand, we need to convert the continuous values into either a one or a zero, but we cannot just round these values based on whether they are higher than 0.5 or not. We need to preserve the ratio of the number of ones to the number of zeros in the augmented dataset, as it was in the original dataset. To do this, for every column, we need to find a threshold, such that, when converting all values below that threshold to zero and all values above it to one, we preserve the ratio split of ones and zeros in the dataset. We implement a hill climbing approach where thresholds starting from 0.1 and incrementing at a step of 0.005 going all the way up to 1 are tested to see which threshold gives the closest split between ones and zeros, when compared to the original dataset. In this manner, all binary columns had their binary state preserved with the newer, augmented values, being converted into binary form after thresholding was done on them to preserve the ratio of ones and zeros.

E. Logistic Regression (LR):

Logistic regression is one of the most commonly used classification algorithms, which is easy to understand and implement as well as provides efficient training. It has been used for heart disease prediction before [14, 15, 16]. It is used to predict binary target variables (0 or 1) from a set of input features. The algorithm uses the sigmoid function to predict a probabilistic value of a target variable between [0,1]. For classifying the binary target, a specific threshold is implemented, which converts these probabilities into binary values (0,1) depending on the comparison between the predicted and threshold value.

- Logistic Sigmoid Function:

p(x) = 11 + e-x where x is the algorithm input, p(x) is the predicted probability and e is the ruler's number equal to 2.71828.

- Cost function used in the logistic regression:

min Ci=1n(-y(i)p(xi)-(1-y(i)(1-(p(xi)) + r where y is the actual target, p(x) is the predicated target and r is the regularization term.

For our dataset, we had to predict the stenosis in three individual arteries (LCA, LCX, RCA) using binary target variable {Yes:1, No:0}. After the data augmentation and pre-processing, the three training datasets were individually fit with the logistic regression model, and accuracy was evaluated on the test dataset. Multiple models were implemented, and finally, the best model performance was chosen for each artery which is described in the results section. Figure 4 shows the architecture of LR.

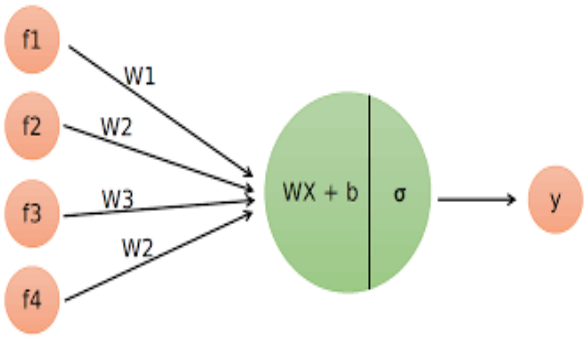

Figure 4. Logistic Regression architecture [17]

F. Random Forest (RF):

The Random Forest algorithm, introduced by Breiman, is a highly effective and most frequently used model, which can be used for classification and regression problems at the same time [18]. It belongs to an ensemble learning technology based on bagging. Its basic idea is to train a set of base classifiers, usually a decision tree, and then aggregate the results of the base classifiers by hard voting or weighted voting to obtain the final prediction output. Therefore, Random Forest usually performs better than a single classifier. In addition, to improve the performance of Random Forest, some strategies need to be adopted, such as the introduction of a greater randomness which can make base classifiers as independent as possible during the process of creating forests. In view of these superiorities, the Random Forest algorithm has been widely used in disease prediction and system development.

In our project, we tried 2,3,4,5,6, and 10 for max_depth parameter in all datasets, and we have got the higher accuracy with 10 in both original dataset and augmented datasets. The max_depth of a tree in Random Forest is defined as the longest path between the root node and the leaf node, and by using the max_depth parameter, I can limit up to what depth I want every tree in my random forest to grow [19]. Figure 5 shows the architecture of RF.

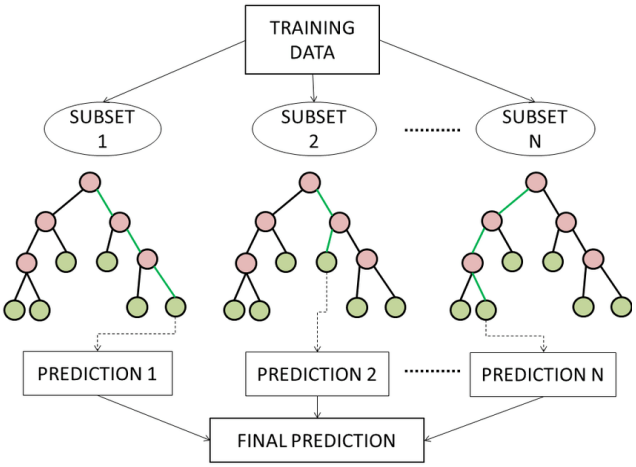

Figure 5. Random Forest architecture [20]

G. Extreme Gradient Boosting (XGBOOST):

Gradient Boosting is a boosting method where errors are minimized using a gradient descent algorithm. Simply put, Gradient descent is an iterative optimization algorithm used to minimize a loss function. The loss function quantifies how

far off our prediction is from the actual result for a given data point. The better the predictions, the lower will be the output of your loss function. [21]

$$MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \hat{y}\right)}_{\text{The square of the difference between actual and predicted}}^{2}$$

XGBoost is an optimized implementation of gradient boosting. Different from Random Forest, the base classifier of XGBoost is interrelated, and the base classifier of the latter is generated based on the former. Specifically, the latter base classifier fits the prediction residuals of the previous base classifier. Based on this integrated strategy, machine learning techniques have shown high performance in solving various disease prediction and risk stratification tasks in recent years [22].

In this project, we applied two hyperparameter tuning for XGBoost, learning_rate, and max_depth. The maximum depth of the individual classification estimators. The maximum depth limits the number of nodes in the tree. For this project, we applied 2,3,4,5,6, and 10 for max_depth parameter in all datasets, and we have got the higher accuracy with 10 in both original dataset and augmented datasets. Learning rate shrinks the contribution of each tree, and we have got the best results in 0.1 for learning_rate.

H. Artificial Neural Networks (ANNs):

Artificial neural networks are an algorithm that mimics the structure of a human brain and tries to recognize the relationship between a large amount of data. They tend to resemble the connection of neurons and synapses found in the human brain. A neural network contains layers of interconnected nodes. These nodes are known as a perceptron, and each of them functions like a multiple linear regression model. The perceptron feeds the signal produced by a multiple linear regression model into an activation function that might not be linear. The input layer collects input patterns. The output layer has classifications or output signals to which input patterns may map [23].

Hidden layers fine-tune the input weightings until the neural network's margin of error is minimal. It is hypothesized that hidden layers extrapolate salient features in the input data that have predictive power regarding the outputs. The accuracy of a neural network greatly depends on the architecture of the various layers that are part of it. The neural network keeps training and revising the weights of each of its nodes till it optimizes a certain loss function. The Loss function that is used by us is the log-loss function. The formula to calculate the log loss functions is:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

In our project, we tried to implement a simple multilayer perceptron classifier to determine whether a person will be suffering from coronary heart disease or not. Different MLP classifiers were trained for every single artery. This was done because the same neural network architecture might not work the best for different arteries. Each artery had a separate neural network trained on it, and a person is said to suffer from coronary heart disease even if one of the arteries is said to be stenotic. We tried various architectures of neural networks for all 3 arteries and found "relu" to be the most effective activation function, while "adam" solver proved the best to converge onto a solution the fastest. Our neural network had 74 nodes in the input layer, one for each feature of the augmented dataset, and only one output that tells us whether the artery is stenotic or not. Figure 6 shows a ANNs schematic.

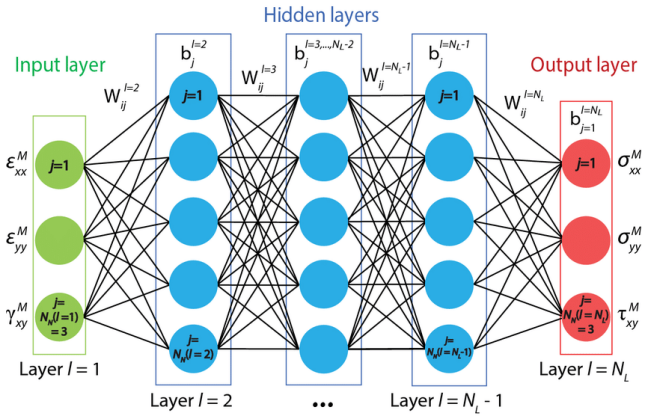

Figure 6. A ANNs schematic [24]

III. EXPERIMENTS AND RESULTS

Our study used the Z-Alizadeh Sani CAD dataset, it is in UCI Machine Learning Repository [4]. In this dataset, we have three outputs for each artery: LAD, LCX, and RCA. The dataset has 303 rows with 55 features as predictors and also three features as target (LAD, LCX, and RCA).

The dataset has four categories: demographical features, symptoms and examination features, ECG features, and laboratory and echocardiography features. As we said, the dataset has three targets including LAD, LCX, and RCA. Hence, we performed the application of Data Augmentation and ML techniques on all of these features.

A. First Experiment(original dataset)

In this step, were applied ML methods on the original dataset with no changes. The results are presented in TABLES I, II, and III.

TABLE I. Comparing the ML Accuracy for **LAD**

Method	Accuracy	F1-Score	Architecture
RF	76.92%	81.08%	max_depth=10
XGBoost	73.62%	76.92%	learning_rate=1.0, max_depth=10
LR	71.42%	74.5%	max_iter :10000, tol: 1e-6, c=50
ANNs	81.31%	85.21%	300

TABLE II. Comparing the ML Accuracy for **LCX**

Method	Accuracy	F1-Score	Architecture
RF	64.83%	42.85%	max_depth=10
XGBoost	65.93%	55.07%	learning_rate=1.0, max_depth=10
LR	57.14%	50.63%	max_iter :10000, tol: 1e-6, c=20
ANNs	63.73%	57.14%	100

TABLE III. Comparing the ML Accuracy for **RCA**

Method	Accuracy	F1-Score	Architecture
RF	63.73%	40%	max_depth=10
XGBoost	61.53%	44.44%	learning_rate=1.0, max_depth=10
LR	65.93%	56.33%	max_iter :10000, tol: 1e-6, c=30
ANNs	65.93%	50.79%	300

The gained results demonstrate that ANNs had the highest accuracy for LAD with 81.31%. XGBoost had best performance for LCX with 65.93%, and LR and ANNs had better performance for RCA with 65.93%.

B. Second Experiment(3000 AEs dataset)

In the section two, we first applied AEs method for data augmentation, and generated 3000 datasets. Moreover, we used thresholding operation and one-hot-encoding for categorical features. After all these steps, we applied ML methods on this new generated dataset with 3000 rows. The results are presented in TABLES IV, V, and VI.

TABLE IV. Comparing the ML Accuracy for **LAD**

Method	Accuracy	F1-Score	Architecture
RF	91.37%	93.03%	max_depth=10
XGBoost	89.93%	91.74%	learning_rate=1.0, max_depth=10
LR	82.63%	85.74%	max_iter :10000, tol: 1e-6, c=30
ANNs	88.49%	90.59%	100,100

TABLE V. Comparing the ML Accuracy for **LCX**

Method	Accuracy	F1-Score	Architecture
RF	91.59%	88.75%	max_depth=10
XGBoost	93.03%	91.16%	learning_rate=1.0, max_depth=10
LR	77.98%	69.14%	max_iter :10000, tol: 1e-6, c=20
ANNs	91.92%	89.95%	100,100

TABLE VI. Comparing the ML Accuracy for **RCA**

Method	Accuracy	F1-Score	Architecture
RF	91.15%	87.13%	max_depth=10
XGBoost	91.81%	88.47%	learning_rate=1.0, max_depth=10
LR	86.61%	81.24%	max_iter :10000, tol: 1e-6, c=100
ANNs	89.16%	84.83%	100,100

The obtained results show that RF had the highest accuracy for LAD of 91.37%. XGBoost showed the best performance for LCX and RCA with an accuracy score of 93.03% and 91.81% respectively.

C. Third Experiment(6000 AEs dataset)

In the third experiment, as same as the second experiment, we first applied AEs method for data augmentation, and generated 6000 datasets. Moreover, we used thresholding operation and one-hot-encoding for categorical features. After all these steps, we applied ML methods on this new generated dataset with 6000 rows. The results are presented in TABLES VII, VIII, and IX.

TABLE VII. Comparing the ML Accuracy for **LAD**

Method	Accuracy	F1-Score	Architecture
RF	87.38%	89.43%	max_depth=10
XGBoost	88.43%	90.06%	learning_rate=1.0, max_depth=10
LR	76.2%	80.41%	max_iter :10000, tol: 1e-6, c=30
ANNs	86.27%	88.07%	100,100,100

TABLE V. Comparing the ML Accuracy for **LCX**

Method	Accuracy	F1-Score	Architecture
RF	91.80%	89.05%	max_depth=10
XGBoost	94.35%	92.84%	learning_rate=1.0, max_depth=10
LR	78.8%	71.48%	max_iter :10000, tol: 1e-6, c=100
ANNs	91.42%	89.07%	150,100,150

TABLE VI. Comparing the ML Accuracy for **RCA**

Method	Accuracy	F1-Score	Architecture
RF	93.74%	91.79%	max_depth=10
XGBoost	94.19%	92.6%	learning_rate=1.0, max_depth=10
LR	79.52%	71.84%	max_iter :10000, tol: 1e-6, c=30
ANNs	92.47%	90.18%	150,150

The results show that XGBoost had the accurate prediction for LAD, LCX, and RCA with 88.43%, 94.35%, and 94.19%, respectively.

D. Fourth Experiment(7500 AEs dataset)

In the fourth experiment, we applied all steps in experiments two and three for generating data with 7500 rows. After all these steps, we applied ML methods on this new generated dataset with 7500 rows. The results are presented in TABLES X, XI, XII.

TABLE X. Comparing the ML Accuracy for **LAD**

Method	Accuracy	F1-Score	Architecture
RF	87.82%	89.68%	max_depth=10
XGBoost	88.84%	90.51%	learning_rate=1.0, max_depth=10
LR	74.32%	79.89%	max_iter :10000, tol: 1e-6, c=50
ANNs	86.32%	88.29%	200,200

TABLE XI. Comparing the ML Accuracy for **LCX**

Method	Accuracy	F1-Score	Architecture
RF	88.84%	83.78%	max_depth=10
XGBoost	93.93%	92.08%	learning_rate=1.0, max_depth=10
LR	79.37%	71.44%	max_iter :10000, tol: 1e-6, c=20
ANNs	91.67%	88.79%	300

TABLE XII. Comparing the ML Accuracy for **RCA**

Method	Accuracy	F1-Score	Architecture
RF	93.40%	90.86%	max_depth=10
XGBoost	93.49%	91.25%	learning_rate=1.0, max_depth=10
LR	80.21%	70.37%	max_iter :10000, tol: 1e-6, c=10
ANNs	91.54%	88.59%	300,300

The results show that XGBoost had the best performance for LAD, LCX, and RCA with 88.84%, 93.93%, and 93.49%, respectively.

E. Fifth Experiment(SMOTE dataset)

In the fifth experiment, we first applied SMOTE method for data augmentation, and generated 423 datasets. Moreover, we used thresholding operation and one-hot-encoding for categorical features. After all these steps, we applied ML methods on this new generated dataset with 423 rows. The results are presented in TABLES XIII, XIV, and XV.

TABLE XIII. Comparing the ML Accuracy for **LAD**

Method	Accuracy	F1-Score	Architecture
RF	86.15%	83.01%	max_depth=10
XGBoost	71.53%	65.42%	learning_rate=1.0, max_depth=10
LR	81.53%	76.47%	max_iter :10000, tol: 1e-6, c=100
ANNs	83.84%	80.37%	300,300

TABLE XIV. Comparing the ML Accuracy for **LCX**

Method	Accuracy	F1-Score	Architecture
RF	83.84%	58.82%	max_depth=10
XGBoost	69.23%	41.1%	learning_rate=1.0, max_depth=10
LR	77.69%	59.15%	max_iter :10000, tol: 1e-6, c=20
ANNs	80.76%	65.75%	100

TABLE XV. Comparing the ML Accuracy for **RCA**

Method	Accuracy	F1-Score	Architecture
RF	74.61%	40%	max_depth=10
XGBoost	69.23%	42.85%	learning_rate=1.0, max_depth=10
LR	73.84%	48.48%	max_iter :10000, tol: 1e-6, c=5
ANNs	77.69%	53.96%	100

The results show that for LAD and LCX the RF had the best performance with 86.15% and 83.84% accuracy, and in RCA the ANNs had best performance with 77.69% accuracy.

F. Sixth experiment (Comparing all datasets and their accuracy results)

In the sixth experiment, we compared all data augmentation methods, and all ML models' together for each artery. The results are presented in TABLES XVI, XVII, XVIII.

TABLE XVI. COMPARISON BETWEEN the ACCURACY of AEs, SMOTE, and ML FOR **LAD**

Method	303 Original	3000 AEs	6000 AEs	7500 AEs	423 SMOTE
RF	76.92%	91.37%	87.38%	87.82%	86.15%
XGBoost	73.62%	89.93%	88.43%	88.84%	71.53%
LR	71.42%	82.63%	76.2%	74.32%	81.53%
ANNs	81.31%	88.49%	86.27%	86.32%	83.84%

The obtained results demonstrate that augmented datasets had mere accuracy than original data, and also, AEs method worked better than SMOTE for data augmentation. Moreover, for LAD in 3000 AEs the RF had the best performance with 91.37% accuracy.

TABLE XVII. COMPARISON BETWEEN the ACCURACY of AEs, SMOTE, and ML FOR **LCX**

Method	303 Original	3000 AEs	6000 AEs	7500 AEs	423 SMOTE
RF	64.83%	91.59%	91.80%	88.84%	83.84%
XGBoost	65.93%	93.03%	94.35%	93.93%	69.23%
LR	57.14%	77.98%	78.8%	79.37%	77.69%
ANNs	63.73%	91.92%	91.42%	91.67%	80.76%

The results show that for LCX, augmented datasets had better performance than original data, and AEs worked better than SMOTE in our dataset. Also, XGBoost had the best accuracy with 94.35% in 6000 datasets.

TABLE XVIII. COMPARISON BETWEEN the ACCURACY of AEs, SMOTE, and ML FOR **RCA**

Method	303 Original	3000 AEs	6000 AEs	7500 AEs	423 SMOTE
RF	63.73%	91.15%	93.74%	93.40%	74.61%
XGBoost	61.53%	91.81%	94.19%	93.49%	69.23%
LR	65.93%	86.61%	79.52%	80.21%	73.84%
ANNs	65.93%	89.16%	92.47%	91.54%	77.69%

The table shows that for RCA, augmented datasets had better performance than original data, and AEs worked better than SMOTE in our dataset. Also, XGBoost had higher accuracy in 6000 datasets with 94.19% accuracy.

IV. DISCUSSION

In the experiment one to experiment five, we discussed the results of our proposed framework. This section demonstrates the results of proposed methodology and compares with other results in the literature to show the power of it. The result is presented in Fig. 7 to Fig. 9.

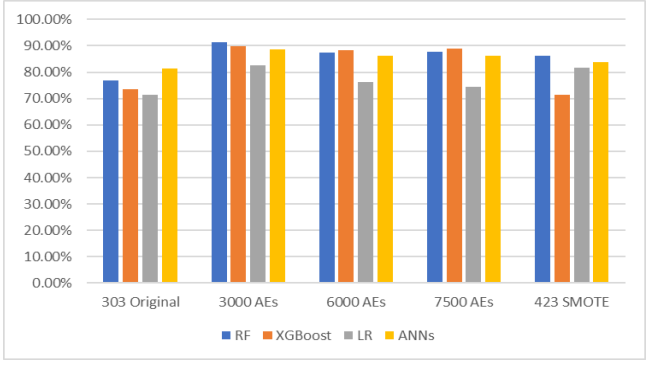

Figure 7. Performance of proposed framework for **LAD**

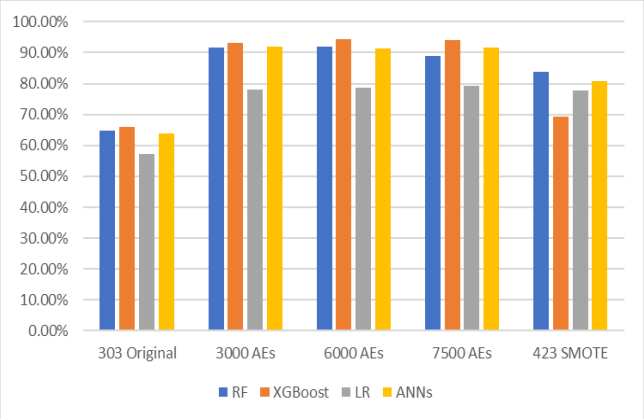

Figure 8. Performance of proposed framework for **LCX**

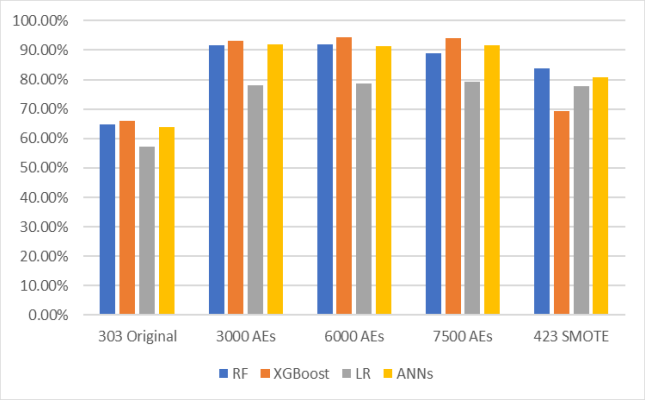

Figure 9. Performance of proposed framework for **RCA**

According to Fig. 7, Fig. 8, and Fig. 9, it can be observed that size of data plays a significant role in order to have better prediction results. For more clarity, the results should be compared with previous studies those applied their methods on the same dataset. Hence, TABLE XIX indicates the comparison of our outcomes with prior results.

TABLE XIX. COMPARISON BETWEEN OUR RESULTS AND OTHER STUDIES

Study	Method	Accuracy
Alizadehsani et al. [25]	Hybrid genetic-discretized algorithm	82.64/ LAD 83.67/ LCX 86.43/ RCA
Alizadehsani et al. [9]	Model uncertainty quantification	86.64/ LAD 83.47/ LCX 82.85/ RCA
Alizadehsani et al. [26]	Combined information gain / SVM	82.64/ LAD 83.67/ LCX 86.43/ RCA
Alizadehsani et al. [3]	Bagging-C4.5	79.54/ LAD 61.46/ LCX 68.96/ RCA
Alizadehsani et al. [27]	C4.5	74.20/ LAD 63.76/ LCX 68.33/ RCA
This Study	ML + Data Augmentation	91.37/ LAD 94.35/ LCX 94.19/ RCA

As TABLE XIX shows, the obtained outcomes in this research are much better than previous results. Moreover, in this study we applied augmentation methods for increasing the size of data, that means, our data was bigger than the previous studies, and as results demonstrate, with more data, we can get higher accuracy.

V. CONCLUSION

This research demonstrated the performance of four classical ML methods: RF, XGBoost, Logistic Regression, and ANNs on the CAD dataset. Three major coronary arteries (LAD, LCX, and RCA) were selected. In other words, we applied the methods on each artery separately. Due to data scarcity, a new framework with two augmentation methods for increasing the size of data, AEs, and SMOTE, was proposed in order to improve the quality of the prediction. Our findings showed that the ML coupled with AEs can improve the prediction outcomes. The highest accuracies were obtained using RF and XGboost approach for LAD, LCX, and RCA which were 91.37%, 94.35 %, and 94.19 %, respectively. As a conclusion, this study introduced a new framework that can outperform previous algorithms having much better results on CAD dataset.

In future works, we want to apply implementing feature reduction techniques like PCA, new augmentation methods, feature selection, and deep learning algorithms.

REFERENCES

1. https://www.cdc.gov/heartdisease/facts.htm
2. Russell MW, Huse DM, Drowns S, Hamel EC, Hartz SC. Direct medical costs of coronary artery disease in the United States. Am J Cardiol. 1998 May 1;81(9):1110-5. doi: 10.1016/s0002-9149(98)00136-2. PMID: 9605051.
3. R., Alizadehsani, J., Habibi, M. J., Hosseini, H., Mashayekhi, R., Boghrati, A., Ghandeharioun,... & Z. A., Sani, " A data mining approach for diagnosis of coronary artery disease", Computer methods and programs in biomedicine, 111(1), 52-61, 2013.
4. Extention of Z-Alizadeh sani dataset Data Set, https://archive.ics.uci.edu/ml/datasets/extention+of+ZAlizadeh+sani+dataset. [accessed on November 5, 2018].
5. Abdar, M., Acharya, U.R., Sarrafzadegan, N. and Makarenkov, V., 2019. NE-nu-SVC: a new nested ensemble clinical decision support system for effective diagnosis of coronary artery disease. IEEE Access, 7, pp.167605-167620.
6. Aouabed, Z., Abdar, M., Tahiri, N., Champagne Gareau, J. and Makarenkov, V., 2019, November. A novel effective ensemble model for early detection of coronary artery disease. In International Conference Europe Middle East & North Africa Information Systems and Technologies to Support Learning (pp. 480- 489). Springer, Cham.
7. Tama, B.A., Im, S. and Lee, S., 2020. Improving an intelligent detection system for coronary heart disease using a two-tier classifier ensemble. BioMed Research International, 2020.
8. Ghiasi, M.M., Zendehboudi, S. and Mohsenipour, A.A., 2020. Decision tree-based diagnosis of coronary artery disease: CART model. Computer methods and programs in biomedicine, 192, p.105400.

9. Alizadehsani, R., Roshanzamir, M., Abdar, M. *et al.* Model uncertainty quantification for diagnosis of each main coronary artery stenosis. *Soft Comput* **24**, 10149–10160 (2020). https://doi.org/10.1007/s00500-019-04531-0
10. Nasarian, E., Abdar, M., Fahami, M.A., Alizadehsani, R., Hussain, S., Basiri, M.E., Zomorodi-Moghadam, M., Zhou, X., Pławiak, P., Acharya, U.R. and Tan, R.S., 2020. Association between work-related features and coronary artery disease: A heterogeneous hybrid feature selection integrated with balancing approach. Pattern Recognition Letters, 133, pp.33-40.
11. Khozeimeh, F., Sharifrazi, D., Izadi, N.H. *et al.* Combining a convolutional neural network with autoencoders to predict the survival chance of COVID-19 patients. *Sci Rep* **11**, 15343 (2021). https://doi.org/10.1038/s41598-021-93543-8
12. SMOTE: Synthetic Minority Oversampling Technique, N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer (2002) https://doi.org/10.48550/arXiv.1106.1813
13. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FIllustration-of-the-SMOTE-oversampling-approach_fig3_347937180&psig=AOvVaw1Kp_tPAEtM4lWM1K5K4qqN&ust=1670461800010000&source=images&cd=vfe&ved=0CBAQjhxqFwoTCKimnqep5vsCFQAAAAAdAAAAABAS
14. Nishadi, A. S. Thanuja. "Predicting Heart Diseases In Logistic Regression Of Machine Learning Algorithms By Python Jupyterlab." (2019).
15. A. Khemphila and V. Boonjing, "Comparing performances of logistic regression, decision trees, and neural networks for classifying heart disease patients," *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, 2010, pp. 193-198, doi: 10.1109/CISIM.2010.5643666.
16. Ambrish G, Bharathi Ganesh, Anitha Ganesh, Chetana Srinivas, Dhanraj, Kiran Mensinkal, Logistic regression technique for prediction of cardiovascular disease (2022)
17. https://www.google.com/url?sa=i&url=https%3A%2F%2Ftowardsdatascience.com%2Fa-logistic-regression-from-scratch-3824468b1f88&psig=AOvVaw1Gl3iKpt2m3EusJ538zgKp&ust=1670462367171000&source=images&cd=vfe&ved=2ahUKEwjXw6ixq-b7AhUzEmIAHU5jCe4Qr4kDegUIARDBAQ
18. Tyralis H, Papacharalampous G, Langousis A. A Brief Review of Random Forests for Water Scientists and Practitioners and Their Recent History in Water Resources. *Water*. 2019; 11(5):910. https://doi.org/10.3390/w11050910.
19. https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
20. https://www.researchgate.net/figure/Example-of-a-Random-Forest-workflow_fig2_342028855
21. https://bradleyboehmke.github.io/HOML/gbm.html
22. https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/
23. https://scikit-learn.org/stable/modules/neural_networks_supervised.html
24. https://www.researchgate.net/figure/Illustration-of-an-FFNN-with-multiple-hidden-layers-NLdocumentclass12ptminimal_fig2_333317530
25. Alizadehsani, R, Roshanzamir, M, Abdar, M, et al. Hybrid genetic-discretized algorithm to handle data uncertainty in diagnosing stenosis of coronary arteries. *Expert Systems*. 2022; 39:e12573. https://doi.org/10.1111/exsy.12573
26. R., Alizadehsani, M. H., Zangooei, M. J., Hosseini, J., Habibi, A., Khosravi, M.,Roshanzamir, ... & S., Nahavandi, "Coronary artery disease detection using computational intelligence methods". Knowledge-Based Systems, 109, 187-197, 2016.
27. Alizadehsani, R., Habibi, J., Bahadorian, B., Mashayekhi, H., Ghandeharioun, A., Boghrati, R., & Sani, Z. A. "Diagnosis of coronary arteries stenosis using data mining". Journal of medical signals and sensors, 2(3), 153, 2012.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper may result in your paper not being published

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your