Atherosclerosis and Atrial Fibrillation Risk Prediction: A Comparative Analysis of Machine Learning Algorithms

Dayanandh Jangamsrinivas   
*dept. of Computer Sciene*  
Central Michigan University *Mount Pleasant, USA*  
[janga1d@cmich.edu](mailto:janga1d@cmich.edu)

Venkata Sri Sai Surya Chalasani  
*dept. of Computer Science*  
Central Michigan University *Mount Pleasant, USA*   
[chala2v@cmich.edu](mailto:chala2v@cmich.edu)

*Abstract*— Atherosclerosis and atrial fibrillation (AF) are two significant cardiovascular disease that have been causing huge number of mortalities. Machine learning (ML) algorithms have shown potential in predicting cardiovascular illnesses by studying large datasets. In this work, we used clinical and demographic data to conduct a complete comparison analysis of several ML algorithms for predicting the risk of atherosclerosis and AF. Our findings show that machine learning algorithms can accurately forecast the risk of both illnesses. Furthermore, we discovered that different algorithms perform differently in predicting the risk of atherosclerosis and AF. The findings of this study imply that machine learning algorithms can considerably enhance risk prediction and aid in the prevention of cardiovascular illnesses. The comparison of the ML algorithms reveals important insights into the creation of precise and economical risk prediction models for atherosclerosis and AF.

Keywords—Atrial Fibrillation, Coronary Artery Disease, Atherosclerosis, Machine Learning, Random Forest, ExtraTree, Voting Test, XGBoost, Decision Tree, K Nearest Neighbors

# Introduction

Atherosclerosis and Coronary Artery Disease (CAD) are primary causes of cardiovascular disease and account for a considerable number of fatalities worldwide [5]. Traditional CAD risk assessment approaches, such as the Framingham Risk Score, are frequently erroneous and rely on a small number of risk variables, such as age, gender, blood pressure, and cholesterol levels [9]. Machine learning and data mining techniques have been widely used in healthcare to solve this issue, notably for predicting cardiovascular disease. In this work, we used clinical and demographic data to predict the risk of atherosclerosis and CAD using several classification algorithms such as K Nearest Neighbors, Random Forest, and Voting Test Classifier [4]. In addition, we created a user-friendly web tool that allows healthcare providers to enter patient data and receive forecasts for atherosclerosis and CAD. The application's user-friendly design makes it simple to access the prediction models used in this study, which can assist physicians in identifying high-risk people and taking appropriate action.

Another key factor to cardiovascular illness is atrial fibrillation (AF), a form of arrhythmia produced by abnormal electrical impulses in the atria [11]. AF can result in insufficient blood flow and a higher risk of blood clots, stroke, heart failure, and other problems. Age, high blood pressure, heart illness, thyroid issues, obesity, sleep apnea, alcohol use, and certain medicines or medical procedures can all be risk factors for AF [12]. Palpitations, tiredness, shortness of breath, dizziness, and chest pain or discomfort are among symptoms of AF.

# Background

The Cleveland dataset has 303 occurrences and 14 characteristics. It is widely used by experts and offers relevant materials. The results show that the K-nearest neighbor has the highest accuracy score. AdaBoost and other ensemble algorithms are successful in improving the prediction performance of poor learners and identifying the risk of cardiovascular disease [8].

To locate the k closest neighbors, samples may be represented as position vectors in a multidimensional feature space, and distance metrics are used to calculate the distance between the samples and the neighbors. The most often used distance metric for kNN is Euclidean distance, which is the metric utilized in this study. We also used a k value of 10 [2].

XgBoost is a well-known supervised learning algorithm based on decision trees that excels in terms of speed and overall performance. Another ensemble strategy, similar to Random Forest, is the boosting approach, which is based on the gradient descent technique to avoid loss when new models are introduced [3].

Decision trees are an effective method for forecasting the onset of coronary heart disease. They can assist physicians in making accurate and timely assessments of a patient's risk of developing coronary heart disease by identifying important risk factors such as age, gender, and lifestyle. Decision trees are simple to understand and may combine a plethora of data to find minor relationships that a physician may not see. They are also good at data classification and prediction [5].

The proposed work is written in Python 3.7.6 and uses the sklearn library, pandas, and other essential tools. This helps to train the model and improve the accuracy of the applied algorithm. Random Forest achieved an accuracy of 81.67%. Decision Tree reaches around 73.33%. Finally, KNN achieved 61.67% accuracy [5].

To construct a new website using Flask MVC, we just use a project generator. The project generator will produce directories and files in a systematic and automated way, eliminating the need for users to create folders and files manually. The project generator has only one input: the name of the project that we will build [10].

To identify characteristics and create prediction models for the beginning of AFF, we used stratified 10-fold cross validation. During the 10-fold cross validation process, the data was divided into 9/10th and 1/10th fractions, which were labeled as the training and testing folds, respectively. The training fold was subjected to feature selection methods, and the ML model was developed utilizing the selected subset of features. The trained ML model is then used to evaluate performance on the testing fold. This procedure was repeated ten times [11].

Atrial fibrillation is a cardiac condition in which irregular heartbeats (arrhythmia) cause blood clots, stroke, cardiovascular failure, and other heart-related issues. It is a frequent form of supportive cardiac arrhythmia that occurs when a high number of electrical signals basically seize control of the pulse, causing the upper chambers of the heart (the atria) to beat rapidly for 400 or more beats per minute and fibrillate [12].

The proposed ML models were trained on a dataset of 23 ECG records with a length of approximately 10 hours each using the leave one group out cross validation (LOGO-CV) technique and achieved the best sensitivity (Se), specificity (Sp), positive predictive value (PPV), false positive rate (FPR), and F1-score of 85.67%, 81.25%, 90.85%, 18.75%, and 88.18%, respectively, to distinguish AF from normal sinus rhythms (NSR) [13].

Because of its excellent performance on a large dataset, XGBoost has been frequently employed and is a well-known classifier for the identification of AF. It is extremely adaptable and can solve most of the regression, classification, and ranking issues. To train the classifier for XGBoost, we utilized the skicit-learn package [14].

# Dataset

## Atherosclerosis or coronary artery disease

For our analysis, we used the Cleveland Clinic Foundation dataset, which is available at the University of California, Irvine [1][2][4]. The dataset has 76 properties, only 14 of which are used, including 13 input fields and one output field. The suggested technique utilizes 303 examples from the dataset's 303 patients with no missing data [8]. The class 0 is means healthy, whereas a patient with a value of 1 has atherosclerosis, often known as coronary artery disease (CAD). 80% of the data is utilized to train the model, whereas 20% is used to test the model. This technique assures that the model's performance is correctly evaluated and that it can forecast the risk of atherosclerosis and CAD in patients. Healthcare workers may take preventive actions to avoid the development of certain illnesses, improving patient outcomes and lowering healthcare costs with precise forecasts.

# Table Description automatically generated

A picture containing application

Description automatically generated

Chart

Description automatically generated with low confidence

## Atrial Fibrillation

The PTB-XL dataset is a freely available collection of electrocardiograms (ECG) recordings from patients with atrial fibrillation (AF) and other cardiac disorders [15]. The Physikalisch-Technische Bundesanstalt (PTB) in Germany established it, and it has over 21,000 ECG recordings from over 10,000 people. This dataset is frequently used in research to create and test automated ECG analysis algorithms, such as those for detecting and classifying cardiac arrhythmias like AF. The PTB-XL dataset is available for non-commercial usage on the PhysioNet website [18]. The signal data in this experiment was stored in nay, a proprietary compressed format. In electrocardiography (ECG), there are 12 standard leads for all signals, with each lead representing a unique arrangement of electrodes connected to the patient's body for recording the electrical activity of the heart. The combination of data from various leads allows for the generation of a full picture of the heart's function.

The limb leads are made up of three bipolar leads (I, II, and III) and three augmented unipolar leads (aver, aVL, and aVF). These leads capture the electrical activity of the heart at various angles between two limbs, such as the right arm and left arm or the right arm and left leg. The electrical potential difference between a single electrode and a virtual central position in the heart is measured by the augmented unipolar limb leads, aver, aVL, and aVF. There are also six unipolar precordial leads, known as chest leads, which are V1, V2, V3, V4, V5, and V6. These leads are critical for detecting and diagnosing changes in the electrical activity of the heart in various areas of the heart, such as during a heart attack [17].

* Lead I, II, and III are the standard bipolar limb leads.
* aVR, aVL, and aVF are the augmented unipolar limb leads.
* V1-V6 are the precordial leads.

The values in each column represent the voltage of the ECG signal recorded from that lead at a specific time point. These values can be used to visualize and analyze the ECG waveform, and to develop algorithms for automated ECG analysis, such as arrhythmia detection and classification.

A dataset was used that contains the corresponding general metadata (such as age, gender, weight, and height).

* ritmi: which is annotated by the PTB-XL team and indicates the patient's heart rhythm. This area records the type of arrhythmia or cardiac disease that was detected in the ECG recording, such as atrial fibrillation, ventricular tachycardia, or myocardial infarction.
* age: which indicates the patient's age in years at the time the ECG was taken.
* sex: The gender of the patient, either male or female.
* height: The height of the patient in centimeters.
* weight: The weight of the patient in kilograms.
* nurse: The ID of the nurse who performed the ECG recording.
* site: The ID of the hospital or medical center where the ECG recording was taken.
* device: The type of ECG device used to record the ECG signal.
* heart\_axis: The electrical axis of the heart, as determined by the PTB-XL team.
* validated\_by: Indicates whether the ECG recording has been validated by an expert annotator or algorithm.
* second\_opinion: Indicates whether a second expert opinion was obtained for this ECG recording.
* validated\_by\_human: Indicates whether the validation was done by a human expert.
* pacemaker: Indicates whether the patient has a pacemaker implanted.
* strat\_fold: Indicates the fold or subset of the data used for cross-validation in a particular study or experiment.

These characteristics can be utilized to assess and comprehend the clinical characteristics of the PTB-XL dataset's patients, as well as to construct machine learning models for ECG analysis and diagnosis.

# Data preprocessing

Diagram

Description automatically generated

Removing irrelevant features is a common step in data preparation, which entails converting raw data into a format appropriate for analysis. The new DataFrame is constructed with a smaller number of features by removing the requested columns, making it simpler to evaluate and use for additional processing or modeling. The columns being dropped are: diagnosi, ecg\_id, patient\_id, recording\_date, report, scp\_codes, infarction\_stadium1, infarction\_stadium2, initial\_autogenerated\_report, baseline\_drift, static\_noise, burst\_noise, electrodes\_problems, extra\_beats, filename\_lr, and filename\_hr[18]. For the age, height, and weight columns, the missing values are being filled in with the mean value of the corresponding column. For the nurse, site, and validated\_by columns, the missing values are being replaced with a value of 0. For the heart\_axis and pacemaker columns, the missing values are being replaced with the string 'Missing'.

These are all typical methods for dealing with missing values in a dataset, and the strategy used may vary depending on the type of missing data and the objectives of the research. As missing values might result in biased or erroneous conclusions in subsequent analyses or models, handling missing values is a crucial step in the preparation of data. As a data preprocessing step, the categorical values in the dataset are mapped to numerical values to make them suitable for the machine learning algorithms. Like, the 'ritmi' column, which represents the cardiac rhythm, is mapped to numerical values 0, 1, and 2 for SR (Normal), AF (atrial fibrillation), and VA (various arrhythmias) respectively. Similarly, other categorical columns such as 'second\_opinion', 'validated\_by\_human', 'heart\_axis', 'pacemaker', and 'device' are also mapped to numerical values based on the information provided in the research paper. This step is necessary because machine learning algorithms require numerical inputs for analysis and prediction.

To minimize the amount of the ECG data, the numpy array file was compressed using the gzip program. The original ECG data was loaded from the 'ECG.npy' file using the numpy library, and the data was saved to a compressed file using the np.save() method and the gzip library. The compressed ECG data was imported from the 'compressed\_npy.gz' file using Python's gzip library. The finished numpy array had 6428 layers, 700 rows, and 12 columns. To limit the number of layers to match the number of rows in the demographic data, the array was partitioned, preserving just the first 700 rows. As a result, a final numpy array with 700 rows and 12 columns was created [18].

We merged the preprocessed ECG data with its corresponding label information. To achieve this, we first converted the 3D ECG array to a 2D array and then to a pandas dataframe. We then renamed the columns to represent the different leads and dropped duplicate rows. Next, we merged this dataframe (out\_df) with a label dataframe (label\_df) based on the common columns index and unique\_id respectively and assigned the resulting dataframe to merged\_df. The index and unique\_id columns were subsequently dropped from merged\_df, leaving us with a combined dataframe that included the ECG data and corresponding labels [18].

# Methodology

We used multiple machine learning algorithms to assess our preprocessed datasets to predict the risk of atherosclerosis and atrial fibrillation.

## Random Forest

The random forest classifier is a well-known machine learning technique that is used for classification, regression, and feature selection problems [3]. It is an ensemble learning method that makes predictions by combining numerous decision trees. Each tree in the forest is created by combining a subset of the original data with a random subset of features. Each tree provides a forecast during training, and the final prediction for a new data point is derived by averaging or taking the mode of all the trees in the forest. Random forest classifiers are well-known for their excellent accuracy, resistance to noise and outliers, and capacity to handle high-dimensional data. When compared to a single decision tree, it is also less prone to overfitting. However, the model's interpretability can be difficult, and the algorithm's performance can be sensitive to the hyperparameters used. The random forest classifier has been used successfully in a variety of sectors, including healthcare, finance, and image classification.

## K-Nearest Neighbors

K Nearest Neighbors (KNN) is a machine learning technique that is non-parametric, and instance based. It is used for classification and regression [4]. The prediction for a new data point in KNN is based on the k closest data points in the training set. A distance metric, such as Euclidean distance or Manhattan distance, is used to calculate the distance between the new data point and the existing points. To avoid ties, the value of k is usually an odd number. KNN is a lazy learning algorithm in that it does not learn a model during training but rather saves the complete training dataset. KNN is simple to develop and interpret, and it can solve binary and multiclass classification problems.

d = √ [ (x22 – x11)2 + (y22 –y11)2]

d=∣a1​−b1​∣+⋯+∣aN​−bN​∣

## Voting Test Classifier

A voting classifier is an ensemble learning technique that predicts by combining numerous individual models. Each individual model in the ensemble makes a forecast under this approach, and the final prediction is selected by a majority vote [5]. This method is applicable to both classification and regression tasks. A voting classifier was used in the current study to predict the likelihood of atherosclerosis and coronary artery disease based on demographic and clinical characteristics. The ensemble's individual models were K Nearest Neighbors and Random Forest, Extratree Classifier, Decsion Tree [6], which were chosen for their shown competence in classification tasks. The majority vote of the individual model forecasts was used to make the final prediction [3]. The current work intended to increase the accuracy of atherosclerosis and coronary artery disease prediction by employing a voting classifier approach rather than a single model. Furthermore, because the integration of many models might assist in reducing the flaws of individual models, this strategy has the potential to be more resistant to alterations in the dataset.

Diagram

Description automatically generated

## XGBoost

XGBoost (eXtreme Gradient Boosting) is a well-known machine learning technique that belongs to the gradient boosting method family. Tianqi Chen created it in 2014 and it has since been widely utilized in a variety of areas. XGBoost is a performance and computational speed enhancement to the traditional gradient boosting technique. Each tree in XGBoost is built sequentially, and the method attempts to minimize the loss function (e.g., binary cross-entropy for binary classification tasks) at each iteration. It accomplishes this by introducing new decision trees into the model, with each new tree designed to fix the faults introduced by the prior tree. To reduce overfitting and increase generalization, XGBoost adds regularization components in the goal function. One of XGBoost's primary benefits is its capacity to deal with missing data, which is prevalent in real-world datasets. It also features a flexible structure that allows users to tailor the objective function and assessment metrics to their own requirements. XGBoost has been effectively applied in a variety of fields, including healthcare, finance, and natural language processing, and has won multiple Kaggle contests.

## ExtraTree Classifier

ExtraTreesClassifier is a machine learning algorithm that works in the same way as the Random Forest algorithm. [6] ExtraTreesClassifier, like Random Forest, constructs a huge number of decision trees, but with one major difference: when determining a split point, it selects a random subset of features rather than searching for the greatest feature, as Random Forest does. This method is also known as "extremely randomized trees," hence the term "ExtraTreesClassifier." ExtraTreesClassifier minimizes the variance of the model by randomizing feature selection, which might lead to higher performance on certain types of data.

## Decision Tree

Decision Trees are an effective machine learning approach that may be applied to both classification and regression issues. They function by splitting the data recursively into smaller and smaller subgroups depending on the most relevant attributes, until the subsets are homogenous with regard to the target variable [13]. A splitting criterion is applied at each node of the tree to determine which characteristic to split on and how to divide the data into two or more groups. Gini impurity and information gain are the most widely utilized splitting criterion. Gini impurity quantifies the likelihood of misclassifying a randomly selected element in the subset, whereas information gain quantifies the reduction in entropy (i.e., the amount of uncertainty) following the split [8].

Text, letter

Description automatically generated

Iteratively, the tree is constructed by repeating the splitting operation on each subset until a halting requirement is reached. This stopping criterion might be based on a variety of parameters, including reaching a maximum depth, obtaining a certain number of samples in each leaf node, or achieving a specific degree of purity. The Decision Tree algorithm's ultimate output is a tree-like model in which each internal node represents a feature, and each leaf node represents a class label or a numerical value for regression issues. The model may then be used to generate predictions on fresh, previously unseen data by traversing the tree from root to leaf node depending on the values of the incoming data's characteristics.

Decision trees offer several advantages, like interpretability, and ability to handle both numerical and categorical data. They can, however, suffer from overfitting, especially if the tree gets too deep or complicated. To solve this issue, numerous Decision Tree variations, like random forests, gradient boosting, and XGBoost, have been created, which integrate several trees to increase performance and prevent overfitting.

## Flask Framework

Flask is a Python-based micro web framework. It is characterized as a micro framework because it does not necessitate the usage of any special tools or libraries. It lacks a database abstraction layer, form validation, and other third-party library components. Flask, on the other hand, allows extensions to add application functionality as if it were built into Flask. Because of its simplicity of use and lightweight nature, Flask is a popular choice for developing small to medium-sized web projects. It follows the Model-View-Controller (MVC) architectural model, allowing developers to create scalable and maintainable web applications [10].

# Performance Evaluation

A critical stage in determining the quality of a machine learning model is performance evaluation. We examined the performance of our classification models in this work using several assessment measures such as accuracy, precision, recall, and F1-score [8]. Precision represents the percentage of genuine positive predictions among all positive forecasts, whereas accuracy counts the percentage of correctly categorized occurrences out of all cases. The proportion of genuine positive predictions among all real positive instances is measured by recall, and the F1-score is a harmonic mean of accuracy and recall that offers a balanced evaluation of the model's performance [8].

We employed a 5-fold cross-validation strategy to guarantee that the model's performance is tested on numerous independent sets of data and to limit the danger of overfitting. The dataset is partitioned into five equal parts in this approach, and each part is used once as a validation set, while the remaining four parts are utilized as the training set. The procedure is done five times, with each portion serving as the validation set just once [8].

We present the results of our models on two distinct datasets: atrial fibrillation and atherosclerosis. The findings show that the Random Forest and Voting Classifier models beat the other models on both datasets, with 96.62% and 96.53% accuracy for Atrial Fibrillation, respectively, and 90.16% accuracy for both models for Atherosclerosis.

Our findings indicate that the Random Forest and Voting Classifier models are good alternatives for categorizing datasets containing Atrial Fibrillation and Atherosclerosis. However, the performance of the models varied depending on the evaluation metric used, with some models outperforming others.

**Atherosclerosis / CAD**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **Fscore** |
| **RF** | 90.16 | 90.21 | 90.30 | 90.16 |
| **KNN** | 91.80 | 91.77 | 91.86 | 91.79 |
| **XGB** | 90.16 | 90.63 | 90.46 | 90.16 |
| **VTC** | 91.80 | 91.55 | 87.5 | 91.80 |

**Atrial Fibrillation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **Fscore** |
| **RF** | 96.64 | 96.48 | 96.76 | 96.61 |
| **KNN** | 96.46 | 96.40 | 96.43 | 96.42 |
| **XGB** | 96 | 95.88 | 96.02 | 95.95 |
| **VTC** | 96.53 | 96.42 | 96.57 | 96.50 |

# Conclusion

Our findings show that machine learning algorithms may be able to effectively predict the risk of cardiovascular diseases such as atherosclerosis and atrial fibrillation. Finally, the algorithms performed well, with all reaching high accuracy in predicting atherosclerosis and atrial fibrillation risk. The findings suggest that machine learning might be used in clinical settings to predict and treat cardiovascular disease. The algorithms' effectiveness varied significantly based on the dataset and the job, but we discovered that ensemble approaches like Random Forest, Voting Test, and XGBoost were very successful. Our findings suggest that future research should employ larger and more varied datasets to increase the efficacy of machine learning algorithms for cardiovascular disease prediction. While machine learning algorithms have demonstrated promise in predicting the risk of cardiovascular disease, they should not be used in place of clinical evaluation by medical professionals. In conclusion, our findings demonstrate that machine learning may predict cardiovascular disease risk and highlight the need for more study in this area to improve accuracy and efficacy.

##### References

1. S. Nikan, F. Gwadry-Sridhar and M. Bauer, "Machine Learning Application to Predict the Risk of Coronary Artery Atherosclerosis," 2016 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2016, pp. 34-39, doi: [10.1109/CSCI.2016.0014](https://doi.org/10.1109/CSCI.2016.0014)
2. A. Newaz and S. Muhtadi, "Performance Improvement of Heart Disease Prediction by Identifying Optimal Feature Sets Using Feature Selection Technique," 2021 International Conference on Information Technology (ICIT), Amman, Jordan, 2021, pp. 446-450, doi: [10.1109/ICIT52682.2021.9491739](https://doi.org/10.1109/ICIT52682.2021.9491739)
3. H. Dhiman, R. Kumar and P. Rani, "A Hybrid Model for Early Prediction of Stroke Disease," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-6, doi: [10.1109/ICRITO56286.2022.9965122](https://doi.org/10.1109/ICRITO56286.2022.9965122)
4. O. Terrada, B. Cherradi, A. Raihani and O. Bouattane, "Atherosclerosis disease prediction using Supervised Machine Learning Techniques," 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 2020, pp. 1-5, doi: <10.1109/IRASET48871.2020.9092082>
5. M. S. Gangadhar, K. V. S. Sai, S. H. S. Kumar, K. A. Kumar, M. Kavitha and S. S. Aravinth, "Machine Learning and Deep Learning Techniques on Accurate Risk Prediction of Coronary Heart Disease," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023, pp. 227-232, doi: [10.1109/ICCMC56507.2023.10083756](https://doi.org/10.1109/ICCMC56507.2023.10083756)
6. N. S, V. K, I. B and J. N. Kalshetty, "Heart Disease Prediction Using Artificial Intelligence Ensemble Network," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-6, doi: <10.1109/MysuruCon55714.2022.9972493>
7. S. K. K. L, N. K. G and M. J. A, "Coronary Artery Disease Prediction using Data Mining Techniques," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 2020, pp. 693-697, doi: [10.1109/ICISS49785.2020.9316014](https://doi.org/10.1109/ICISS49785.2020.9316014)
8. K. N. Devi, S. Suruthi and S. Shanthi, "Coronary Artery Disease prediction using Machine Learning Techniques," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 1029-1034, doi: [10.1109/ICACCS54159.2022.9785140](https://doi.org/10.1109/ICACCS54159.2022.9785140)
9. P. M. Brindle, “The accuracy of the Framingham risk-score in different socioeconomic groups: a prospective study,” British Journal of General Practice, Nov. 01, 2005.

<https://bjgp.org/content/55/520/838#:~:text=Recent%20studies%20have%20shown%20that%20Framingham%20risk%20scores,circumstances%20may%20be%20excluded%20from%20receiving%20preventive%20treatment>.

1. M. R. Mufid, A. Basofi, M. U. H. A. Rasyid, I. F. Rochimansyah, and A. Rokhim, *Design an MVC Model using Python for Flask Framework Development*. 2019. doi: [10.1109/ELECSYM.2019.8901656](https://doi.org/10.1109/ELECSYM.2019.8901656)
2. S. Karnik et al., "Predicting atrial fibrillation and flutter using Electronic Health Records," 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, San Diego, CA, USA, 2012, pp. 5562-5565, doi: [10.1109/EMBC.2012.6347254](https://doi.org/10.1109/EMBC.2012.6347254)
3. M. Almazrouei and M. Al-Rajab, "Automated Indicator of Atrial Fibrillations Risk Using Machine Learning," 2021 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2021, pp. 1229-1235, doi: [10.1109/CSCI54926.2021.00254](https://doi.org/10.1109/CSCI54926.2021.00254)
4. M. S. Jahan, M. Mansourvar, S. Puthusserypady, U. K. Wiil, and A. Peimankar, “Short-term atrial fibrillation detection using electrocardiograms: A comparison of machine learning approaches,” *International Journal of Medical Informatics*, vol. 163, p. 104790, May 2022, doi: <https://doi.org/10.1016/j.ijmedinf.2022.104790>
5. S. Liaqat, K. Dashtipour, M. Imran, K. Assaleh, K. Arshad, and N. Ramzan, “Detection of Atrial Fibrillation Using a Machine Learning Approach,” *Information*, vol. 11, no. 12, p. 549, Nov. 2020, doi: [10.3390/info11120549](https://doi.org/10.3390/info11120549)
6. S. R. S K and R. J. Martis, "Machine Learning Based Decision Support System for Atrial Fibrillation Detection using Electrocardiogram," 2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Udupi, India, 2020, pp. 263-266, doi: [10.1109/DISCOVER50404.2020.9278124](https://doi.org/10.1109/DISCOVER50404.2020.9278124)
7. Wagner, Patrick, Nils Strodthoff, Ralf-Dieter Bousseljot, Wojciech Samek, and Tobias Schaeffter. “PTB-XL, a Large Publicly Available Electrocardiography Dataset.” PTB-XL, a large publicly available electrocardiography dataset v1.0.1, April 24, 2020. <https://physionet.org/content/ptb-xl/1.0.1/>
8. “PTB-XL - Atrial Fibrillation Detection,” *Kaggle*, Apr. 16, 2021. <https://www.kaggle.com/datasets/arjunascagnetto/ptbxl-atrial-fibrillation-detection>

**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.**