Arrhythmia Classification Using Machine Learning

Overview

This study introduces a comprehensive framework for the automated analysis of electrocardiogram (ECG) data, addressing the critical need for robust and efficient methods in light of the vast volume of data generated by modern wearable technologies such as the Apple Watch. These devices have transformed healthcare by enabling continuous cardiovascular monitoring and facilitating the early detection of abnormalities. [1]

The project's core objective is to develop and evaluate advanced classification techniques for arrhythmia detection. To achieve this, the framework integrates three key innovations. First, an ensemble model is proposed to combine the strengths of various classifiers, thereby enhancing overall prediction accuracy. This approach seeks to leverage the collective intelligence of multiple models to achieve more robust performance. Second, the framework addresses a notable gap in existing research by incorporating simpler models, such as K-Nearest Neighbors (KNN), Logistic Regression, Naive Bayes, and Decision Trees. These models are often overlooked in advanced ECG analysis despite their inherent interpretability and computational efficiency, offering a different perspective on model suitability. Third, a Convolutional Neural Network (CNN) is employed for image-based recognition of ECG signals. This innovative application facilitates the extraction of subtle morphological features from visual representations of the signals, enabling a pathway for real-time analysis. [1]

The deliberate inclusion of both highly complex, sophisticated models and simpler, more interpretable alternatives underscores a commitment to a holistic model evaluation. This approach extends beyond merely identifying the highest-accuracy model; it provides a nuanced understanding of model suitability for various deployment scenarios, particularly those with practical constraints. This pragmatic

perspective is crucial, as it acknowledges that the most effective solution in a real-world setting may not always be the most computationally intensive one. The proposed multi-model approach is designed to not only improve diagnostic performance but also to offer a scalable solution for effective cardiovascular monitoring in practical healthcare settings, considering factors like resource availability and the need for explainability. [1]

Keywords: Machine Learning, ECG, Arrhythmia, Classification, XGBoost, KNN, Deep Learning, Neural Networks, CNN. [1]

1. Introduction

The advent of connected health technologies embedded in wearable devices, such as Apple Watches, has ushered in a transformative era of continuous cardiovascular monitoring. These devices hold immense potential to revolutionize patient care by enabling early detection and real-time surveillance of heart-related conditions, moving beyond traditional, sporadic diagnostic methods. However, the sheer volume of ECG data generated by these ubiquitous devices presents significant challenges for conventional diagnostic approaches, which often rely on manual interpretation and heuristic feature extraction, proving inefficient for large-scale, continuous data streams. [1]

Recent advancements in machine learning and deep learning have offered promising alternatives for automated ECG analysis. Sophisticated algorithms have been successfully deployed to improve diagnostic accuracy by automatically classifying heartbeats, leveraging complex feature extraction techniques and ensemble strategies to overcome limitations of simpler learning architectures. Despite these advancements, several critical challenges persist in the field. [1]

A primary challenge is the predominant focus in existing studies on advanced, computationally intensive models. While effective, these models often necessitate extensive pre-processing and intricate parameter tuning, frequently overshadowing the potential benefits of incorporating simpler models. These simpler models can offer competitive performance with distinct advantages in interpretability and reduced computational overhead. In clinical settings, understanding the rationale behind a model's diagnosis is often as vital as its accuracy, fostering trust among medical professionals and enabling informed decision-making. The project's aim to elucidate the trade-offs between model simplicity and predictive power directly addresses this need, acknowledging that complex, opaque models, despite high accuracy, may encounter significant adoption barriers due to a lack of explainability. [1]

A second critical challenge lies in the underexplored integration of image recognition techniques into ECG analysis. Traditionally, ECG signals are treated as one-dimensional time-series data. However, transforming these signals into visual representations and applying Convolutional Neural Networks (CNNs) for image-based recognition presents a novel approach. This method could potentially enhance the detection of subtle morphological features within the ECG waveform, leading to improved diagnostic performance and enabling real-time analysis. This methodological innovation could unlock deeper insights by allowing CNNs, powerful architectures typically used for spatial data, to identify intricate patterns that might be difficult to extract using conventional 1D signal processing or hand-crafted feature engineering. This approach is predicated on the belief that visual patterns in ECGs contain rich, yet underexplored, diagnostic information. [1]

In response to these challenges, this study proposes a unique multi-model approach with two primary objectives: [1]

- **Objective 1: Incorporation of Simpler Models:** To provide a more holistic evaluation of ECG analysis methods, the study integrates and compares simpler models, including K-Nearest Neighbors, Logistic Regression, Naive Bayes, and Decision Trees, with more complex methods. This comparison aims to thoroughly examine the trade-offs between model simplicity and predictive power, informing choices for practical deployment. [1]
- **Objective 2: CNN-based Image Recognition:** To enhance the detection of subtle morphological features and enable real-time, automated interpretation of ECG signals, the project leverages CNN architectures to perform image recognition on ECG data. [1]

2. Dataset Overview: MIT-BIH Arrhythmia Database

The study utilizes the **MIT-BIH Arrhythmia Database**, a widely recognized benchmark dataset in the field of cardiac arrhythmia research and for the evaluation of automated ECG analysis algorithms. This comprehensive database was developed by the Massachusetts Institute of Technology (MIT) and Beth Israel Hospital (now Beth Israel Deaconess Medical Center) and has been publicly available since 1980 through PhysioNet. It serves as a standard testbed for various tasks in electrocardiography (ECG) research, including arrhythmia detection, signal processing, and classification. [1]

The database comprises 48 half-hour excerpts of two-channel ambulatory ECG recordings, collected between 1975 and 1979 from 47 subjects, with one subject contributing two separate recordings. The selection of these recordings was meticulous: 23 were randomly chosen from a larger set of 4,000 long-term (24-hour) ECG recordings from both inpatients (approximately 60%) and outpatients (approximately 40%) at Boston's Beth Israel Hospital. To ensure a comprehensive representation of cardiac conditions, an additional 25 recordings were specifically chosen to include less common but clinically significant arrhythmias that would not be adequately represented in a purely random selection. [1]

A hallmark of this dataset is its high-quality annotation. Each heartbeat in the database has been manually annotated by at least two independent cardiologists, and any discrepancies between annotations were resolved through subsequent review, ensuring reliable reference labels for supervised learning approaches. The dataset contains approximately 110,000 annotated heartbeats, each labeled with a corresponding class indicating the type of beat or rhythm abnormality. [1]

The key characteristics and class labels of the dataset are summarized below: [1]

Characteristic	Value
Number of Samples	109,446
Number of Categories	5
Sampling Frequency	125Hz
Classes and Descriptions	

'N': 0	Normal beat
'S': 1	Supraventricular premature beat
'V': 2	Premature ventricular contraction
'F': 3	Fusion of ventricular and normal beat
'Q': 4	Unclassifiable beat

Data Pre-processing Steps

To ensure compatibility and optimal performance with various machine learning and deep learning models, the dataset underwent several crucial pre-processing transformations. First, **downsampling** was applied: the original MIT-BIH dataset, with a sampling rate of 360 Hz (360 data points per second), was reduced to 125 Hz. This step significantly decreases the computational load while effectively preserving essential heartbeat features necessary for classification. Second, **cropping** was performed, where the ECG waveform for each heartbeat was extracted using a fixed window centered around the R-peak. This ensures consistent input sizes for all models, as any extra data beyond the required segment length is removed. Finally, **padding with zeros** was applied to samples shorter than the required length (e.g., 187 time points). This technique ensures that all input samples to a model have a uniform shape, which is a common requirement for many machine learning algorithms. [1]

Insights from Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to gain a deeper understanding of the MIT-BIH Arrhythmia dataset, identify potential patterns, imbalances, and correlations that could influence model performance. A major observation from the EDA was the significant **class imbalance** across the five heartbeat categories. The 'N' (Normal) class is heavily dominant, accounting for over 70,000 instances in the training set and approximately 18,000 in the test set. In stark contrast, the remaining four classes—'S' (Supraventricular ectopic beat), 'V' (Ventricular ectopic beat), 'F' (Fusion beat), and 'Q' (Unknown beat)—are significantly underrepresented, with Class F contributing the fewest samples. [1]

Despite this pronounced imbalance, the study deliberately chose *not* to apply balancing techniques for the simple and sophisticated models. This decision reflects a nuanced understanding of the domain, prioritizing clinical realism over algorithmic idealism. Imbalanced distributions in arrhythmia datasets are not merely an artifact of data collection; they accurately reflect real-world prevalence, where normal heartbeats are naturally far more common than abnormal ones. Artificially altering this distribution through techniques like oversampling or undersampling carries several risks. It can lead to models overfitting to rare classes, potentially learning noise or artifacts rather than genuine patterns, which reduces their generalizability on real-world ECG signals. Furthermore, under-sampling the dominant 'N' class discards valuable data that could aid in distinguishing subtle deviations from normality, a crucial aspect of ECG analysis. Finally, while a balanced dataset might inflate metrics like precision and recall for minority classes during validation, performance could degrade significantly when deployed on naturally imbalanced real-world data, leading to misleading performance assessments. Instead of altering the data distribution, the approach focused on utilizing robust models, particularly Convolutional Neural Networks (CNNs) trained on a down-sampled balanced subset, and meticulously monitoring class-specific performance metrics such as per-class F1-scores. This strategy acknowledges that a model performing well on an artificially balanced dataset might fail catastrophically when deployed in a true clinical setting, necessitating a critical focus on specific metrics for minority classes. [1]

Additionally, a correlation heatmap was plotted to understand the inter-feature relationships within the dataset. This heatmap revealed strong pairwise Pearson correlation coefficients between many features, especially along the diagonal, indicating self-correlation of 1.0. Certain feature clusters exhibited high correlation, suggesting a degree of redundancy among the features. This observation implies that while the raw data is abundant (~110,000 heartbeats), its informational content per feature might be lower than expected due to this redundancy. This redundancy highlights the necessity for dimensionality reduction techniques, such as Principal Component Analysis (PCA), not only for enhancing computational efficiency but also for potentially improving model generalization by removing noise or less informative features. This suggests that simply having "more data" does not automatically translate to "better features," and intelligent data transformation is critical for effective model training. [1]

3. Machine Learning Models Employed

The study systematically evaluates a diverse range of machine learning models, categorized into "Simple Models," "Sophisticated Models," and a "Deep Learning Model." This structured approach facilitates a comprehensive comparison across different paradigms of machine learning, allowing for a nuanced understanding of their performance-complexity landscape. This broad model selection aims to identify not just the highest-performing model, but the most appropriate model given real-world constraints such as interpretability in clinical settings or computational resources on wearable devices. [1]

Simple Models

These models are generally characterized by their lower computational intensity and often offer better interpretability, making their decision-making processes more transparent. [1]

- **K-Nearest Neighbors (KNN):** This is a non-parametric, supervised learning method predominantly used for classification, though it can also be applied to regression tasks. The fundamental principle of KNN is that similar data points tend to exist in close proximity within the feature space. When classifying a new, unseen data point, KNN calculates the distance (typically Euclidean or Manhattan) between this query point and every other point in the training set. It then identifies the 'k' closest examples and assigns the class that appears most frequently among these neighbors. Key hyperparameters that influence KNN's performance include the number of neighbors (k) and the chosen distance metric. [1]

- **Logistic Regression (Multinomial):** This model is a generalization of binary logistic regression, extended for multiclass classification problems. Its objective is to predict the probability that a given instance belongs to one of several distinct classes (e.g., Class 0, Class 1,..., Class K). Unlike binary logistic regression which uses a single sigmoid function, multinomial logistic regression employs the softmax function. This function maps a set of linear scores into a probability distribution over all classes, ensuring that the probabilities for all classes sum to one. [1]
- **Decision Trees:** A Decision Tree Classifier is a supervised learning model that makes predictions by following a tree-like structure. This structure is composed of internal nodes, which represent tests on specific attributes of the data; branches, which correspond to the possible outcomes of these tests; and leaf nodes, which hold the final class labels or predictions. This model is particularly valuable when there is a need to clearly visualize all possible outcomes of a problem and to examine the implications of each decision path, offering high interpretability. [1]
- **Naive Bayes:** Naive Bayes represents a family of simple yet powerful probabilistic classifiers built upon Bayes' Theorem. Its foundational assumption is that features are conditionally independent given the class label, which significantly simplifies computations. Despite this simplifying assumption, Naive Bayes has demonstrated competitive performance across various classification tasks, proving particularly effective when dealing with high-dimensional data and limited training samples. [1]

Sophisticated Models

These models are typically more advanced and often computationally intensive, designed to achieve higher predictive power by capturing more complex patterns in data. [1]

- **XGBoost (Extreme Gradient Boosting):** XGBoost stands as a leading advancement in machine learning, evolving from the Gradient Boosting Regressor Tree (GBRT) technique. It enhances efficiency by simplifying individual regression trees and introducing a highly optimized tree-boosting framework. Recognized for its effectiveness, XGBoost is an ensemble technique that amalgamates multiple weak models to construct a formidable predictive model. A core innovation is its integration of an objective function that combines both a loss function and regularization terms, seamlessly woven into the conventional GBRT loss function. This integration provides XGBoost with exceptional flexibility and adaptability, enabling it to perform with precision across diverse datasets and learning tasks. [1]

- **SVC (Support Vector Classifier) and Linear SVC:** Support Vector Machine (SVM) is a robust machine learning technique used in supervised learning tasks to define an optimal hyperplane that effectively separates data points belonging to different classes. SVM can handle non-linear separability by employing kernel mapping, which transforms feature vectors into higher-dimensional spaces where they may become linearly separable. SVC is a general and powerful classification algorithm. Linear SVC is a variant that, by default, uses a linear kernel. This characteristic makes Linear SVC particularly well-suited for datasets that exhibit linear separability and often offers computational advantages over SVC with a linear kernel, especially when dealing with large-scale datasets, due to its typically faster training time. [1]
- **Random Forest (RF):** Random Forest is a powerful ensemble learning algorithm that enhances predictive accuracy and robustness by combining the predictions of multiple decision trees. It operates based on the principle of bagging (bootstrap aggregating), where each individual tree in the forest is trained on a random subset of the data, sampled with replacement. Further randomness is introduced at each split within a decision tree, where only a random subset of features is considered. This dual source of randomness helps to reduce correlation among the individual trees, leading to improved generalization capabilities and reduced overfitting compared to single decision tree models. [1]
- **LightGBM (Light Gradient Boosting Machine):** Developed by Microsoft, LightGBM is a state-of-the-art machine learning algorithm designed for high performance in both classification and regression tasks. It operates within the gradient boosting framework, but unlike traditional boosting methods that grow trees level-wise, LightGBM grows trees in a leaf-wise manner. This means it focuses on expanding the leaf with the greatest potential to reduce loss, often resulting in faster convergence and higher accuracy. A key innovation is its histogram-based algorithm, which discretizes continuous features into bins, significantly reducing memory usage and improving training speed. LightGBM also natively supports categorical variables, eliminating the need for one-hot

encoding, and is optimized for large-scale data with support for parallel computation and GPU acceleration. [1]

The detailed descriptions of ensemble methods like XGBoost, Random Forest, and LightGBM highlight their ability to combine multiple weaker learners to form a more robust and accurate predictive model. This collective strength is a recurring advantage, suggesting that the inherent complexity and potential noise in medical ECG datasets can benefit significantly from the robustness, reduced overfitting, and enhanced predictive power that these advanced techniques offer. [1]

Deep Learning Model

This category involves neural networks with multiple layers, distinguished by their capacity to automatically extract complex features from raw data, often without explicit feature engineering. [1]

- **CNN (Convolutional Neural Network):** A Convolutional Neural Network (CNN) is a type of feed-forward neural network specifically designed to process spatial data, such as images. CNNs have been extensively applied to various visual tasks, including image classification, and have also shown success in natural language processing, particularly in automatic speech recognition. The typical architecture of a CNN comprises several key components: convolutional layers, which are responsible for capturing local patterns within the input data; ReLU (Rectified Linear Unit) activation layers, which introduce non-linearity; pooling layers, which reduce dimensionality while preserving essential features; and fully connected output layers, which produce the final classification. CNNs are primarily utilized for their ability to extract hierarchical spatial features from data where spatial relationships between data points are significant. [1]

4. Methodology and Experimental Setup

The research followed a meticulously structured pipeline for the classification of cardiovascular conditions using machine learning techniques. This involved distinct workflows tailored for the simple and sophisticated models, and a specialized approach for the Convolutional Neural Network (CNN) due to its image-based processing nature. [1]

Overall Structured Pipeline for Simple and Sophisticated Models

The methodology for the simple and sophisticated models adhered to a comprehensive, multi-stage pipeline: [1]

1. **Data Loading and Exploratory Data Analysis (EDA):** The initial step involved loading the MIT-BIH Arrhythmia dataset into the computational environment. Following this, an extensive EDA was performed to understand the dataset's inherent structure, identify any missing values, and examine key patterns across the different heartbeat classes. A significant observation from this analysis was the pronounced class imbalance, with the 'N' (Normal) class being highly dominant. [1]
2. **Data Cleaning and Preprocessing:** To ensure the data was suitable for model training, cleaning procedures were implemented. Missing values were addressed using mean imputation. Subsequently, normalization techniques were applied to uniformly scale the features. This step is particularly crucial for distance-based models, such as K-Nearest Neighbors, as it prevents features with larger numerical ranges from disproportionately influencing distance calculations. [1]
3. **Data Splitting:** The preprocessed dataset was then partitioned into training and testing sets using a 70:30 ratio. This standard split ensures a reliable evaluation of the models' generalization capabilities on unseen data, providing an objective measure of their real-world performance. [1]
4. **Feature Engineering (Principal Component Analysis - PCA):** To reduce the dimensionality of the dataset and enhance computational efficiency, Principal Component Analysis (PCA) was applied. PCA is an unsupervised linear transformation method designed to transform a high-dimensional dataset into a lower-dimensional space while preserving as much of the data's variability as possible. It identifies patterns in data and expresses them in a way that highlights their similarities and differences by computing the eigenvectors and eigenvalues of the covariance matrix. In this study, the top 30 features that captured the most significant amount of variation in the data were selected for use in model

training. [1]

5. **Model Training:** A diverse set of machine learning models were trained on the prepared data. This included "Simple Models" such as K-Nearest Neighbors (KNN), Logistic Regression, Naive Bayes, and Decision Trees. Additionally, "Sophisticated Models" like XGBoost, Support Vector Classifier (SVC), Linear SVC, and Random Forest were trained to evaluate their respective classification performances. [1]
6. **Model Evaluation:** The performance of the trained models was rigorously evaluated using a suite of standard classification metrics. These included accuracy, precision, recall, and F1-score, which provide comprehensive insights into how well the models predict target classes. The Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) was also utilized to assess the discriminative ability of the models. The report defines these metrics: True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), and provides formulas for Accuracy ((TP+TN)/(TP+TN+FP+FN)), Precision (TP/(TP+FP)), Recall (TP/(TP+FN)), and F1 Score (2*(P*R)/(P+R)). [1]
7. **Model Analysis:** Following evaluation, the optimized model underwent a comprehensive performance analysis using the aforementioned evaluation metrics. This detailed analysis led to the selection of the most effective model for the task of arrhythmia classification. [1]

CNN Approach Workflow

The Convolutional Neural Network (CNN) approach followed a distinct workflow, specifically tailored for image-based analysis of ECG signals, which fundamentally differs from the pipeline used for traditional machine learning models. The use of two distinct methodological pipelines underscores the adaptability required in data science projects; the choice of model profoundly influences the entire data preparation and processing pipeline, emphasizing that there is no one-size-fits-all approach. [1]

1. **Dataset Preparation:** The initial step involved loading and pre-processing the dataset. Subsequently, each dataset was meticulously split into features (X) and their corresponding labels (y). [1]
2. **Class Down-sampling:** To ensure uniform representation across all classes and

to maintain computational efficiency during CNN training, a strategic class down-sampling was performed. Each class was reduced to 400 samples for the training set and 150 samples for the test set. This contrasts with the simple/sophisticated models, which were trained on the full, imbalanced dataset. This pragmatic approach to CNN training, given that CNNs are computationally intensive and can be sensitive to class imbalance, likely enabled faster experimentation and more stable training for the CNN. It allowed the model to learn features from all classes without being overwhelmed by the dominant 'N' class. However, it is important to note that the CNN's reported results are therefore based on a different data distribution than the other models, requiring careful interpretation when comparing overall accuracy. [1]

3. **Signal-to-Image Conversion:** A crucial and innovative step for the CNN approach involved reshaping and transforming the 1D ECG signal samples into 100x100-pixel PNG images. This conversion was essential as it enabled the use of 2D CNN architectures, which are inherently designed to perform well on image data. This novel approach allows the powerful capabilities of 2D image processing architectures to be leveraged for inherently 1D time-series data, potentially uncovering new diagnostic patterns from the visual representations. Examples of these converted images are illustrated in Figure 4 and Figure 5 of the source document. [1]
4. **Data Pipeline Construction:** Pandas DataFrames were created to establish a clear mapping between image filenames and their corresponding class labels. To handle batch processing, shuffling, and real-time data feeding to the model during training and evaluation, ImageDataGenerators were implemented for both the training and testing datasets. [1]
5. **Model Architecture and Compilation:** A custom 2D CNN architecture was meticulously designed, specifically optimized for the task of medical image classification. The model was then compiled using an appropriate optimizer, such as Adam, and a categorical loss function, specifically categorical_crossentropy, which is suitable for multi-class classification problems. [1]
6. **Model Training:** The CNN was trained iteratively, incorporating early stopping mechanisms to prevent overfitting and model checkpointing to save the best-performing model throughout the training process. Training metrics, including loss and accuracy, were continuously monitored and visualized over epochs to track the model's learning progression. [1]
7. **Evaluation:** The trained CNN model underwent a thorough evaluation on the test set. This involved assessing its performance using multiple metrics, including

overall accuracy, a detailed classification report providing precision, recall, and F1-Score per class, and a Confusion Matrix to visually represent class-wise predictions and misclassifications. [1]

8. **Model Saving:** Upon completion of training and evaluation, the final trained CNN model was saved. This allows for future inference and deployment without the need for retraining. [1]

5. Results and Performance Analysis

The performance of all evaluated models was rigorously assessed using a consistent set of classification metrics: Accuracy, Precision, Recall, and F1-Score. These metrics provide a holistic view of each model's effectiveness in predicting the target classes. [1]

5.1 Simple Models

The evaluation of simple models revealed varying levels of performance, highlighting the trade-offs between model simplicity and predictive power. The K-Nearest Neighbors (KNN) algorithm demonstrated the best overall performance among this category, exhibiting a strong capability to distinguish between normal and arrhythmic cases with high reliability. Decision Tree also performed effectively, showing good generalization and interpretability, which is particularly valuable in clinical decision-making duemaking. Logistic Regression achieved moderate results, indicating decent predictive power but lower sensitivity compared to non-linear models. In contrast, Naive Bayes performed poorly, likely because its strong independence assumptions do not hold true in this medically complex dataset, which often involves correlated physiological variables. [1]

The overall performance metrics for these simple models are presented in Table 1. [1]

Table 1: Performance Metrics for Simple Models

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.9162	0.8026	0.7756	0.7926
K-Nearest Neighbours	0.9711	0.9289	0.8535	0.8756
SVC	0.9670	0.8001	0.7565	0.7856
Decision Tree	0.9547	0.8127	0.7999	0.8060

K-Nearest Neighbors (KNN) Detailed Performance

The KNN model demonstrated excellent performance across most arrhythmia classes, achieving an overall accuracy of 97.30%. As expected due to its large support (14,577 instances), Class N (Normal) showed the highest precision (0.9774), recall (0.9933), and F1-score (0.9853). However, minority classes such as S (Supraventricular premature beat) and F (Fusion of ventricular and normal beat) exhibited lower recall values (0.6340 and 0.7039, respectively), indicating a higher rate of false negatives for these clinically concerning classes. Despite these imbalances, the macro average F1-score of 0.8756 reflects a relatively strong ability to generalize across all classes, while the weighted average F1-score of 0.9718 underscores the model's strong overall performance, primarily driven by the dominant class. [1]

The class-wise performance metrics for KNN are detailed in Table 2. [1]

Table 2: KNN Classification Metrics

Class	Precision	Recall	F1-Score	Support

0 (N)	0.9774	0.9933	0.9853	14,577
1 (S)	0.9170	0.6340	0.7496	418
2 (V)	0.9258	0.8804	0.9025	1,120
3 (F)	0.8359	0.7039	0.7643	152
4 (Q)	0.9885	0.9646	0.9764	1,244
Accuracy			**0.9730**	**17,511**
Macro Avg	0.9289	0.8353	0.8756	17,511
Weighted Avg	0.9722	0.9730	0.9718	17,511

The confusion matrix for the KNN classifier further illustrates its predictive performance. It shows strong performance for the majority class 0 (normal rhythm), with 14,480 correct predictions out of 14,577 instances, indicating high reliability for ruling out arrhythmias. However, for arrhythmic classes, the model exhibits notable class confusion. Specifically, 150 instances of Class S were misclassified as Class 0, contributing to its low recall of 63.4%. This is clinically concerning, as false negatives could delay diagnosis and treatment. Similarly, while Class V's recall is relatively high (88.04%), 118 instances were confused with Class 0, which could impact early detection efforts. Class F, a minority class, saw 107 out of 152 instances correctly classified, but with a noticeable spread across predicted classes, particularly to Class 0 and Class 2. Class Q, however, showed excellent performance with 1200 correct predictions out of 1244, reflecting both high recall and precision. The confusion matrix suggests that most misclassifications occur due to overlapping features with the dominant class, indicating that while KNN excels in identifying normal patterns, its sensitivity to rare arrhythmias may require improvement through techniques like resampling or feature engineering. [1]

The KNN model demonstrates a training accuracy of 98.01% and a test accuracy of

97.30%. This small gap between training and test performance indicates excellent generalization with minimal overfitting, suggesting the model performs consistently on unseen data. [1]

Decision Tree Detailed Performance

The Decision Tree model achieved a high overall accuracy of 95.65%, with strong performance on the dominant Class N (F1-score: 0.9765). However, it struggled with minority classes, particularly Class F (F1-score: 0.6348) and Class S (F1-score: 0.6471), indicating that the class imbalance may have affected its ability to generalize effectively to these less represented categories. The weighted average F1-score of 0.9563 demonstrates overall reliable predictions when weighted by class frequency. [1]

The class-wise performance metrics for the Decision Tree are detailed in Table 3. [1]

Table 3: Decision Tree Classification Metrics

Class	Precision	Recall	F1-Score	Support
0 (N)	0.9763	0.9767	0.9765	14,577
1 (S)	0.6633	0.6316	0.6471	418
2 (V)	0.8616	0.8723	0.8669	1,120
3 (F)	0.6187	0.6218	0.6348	152
4 (Q)	0.9435	0.9534	0.9484	1,244
Accuracy			**0.9565**	**17,511**
Macro Avg	0.8127	0.7999	0.8060	17,511

Weighted Avg	0.9560	0.9565	0.9563	17,511

The confusion matrix for the Decision Tree classifier confirms its strong performance on Class N, with 14,237 correct predictions. However, it also clearly illustrates its challenges with minority classes; Class S, for example, is frequently confused with Class N (147 instances misclassified). While Class V shows some overlap with Classes N and S, Class Q is well-identified with 1,186 correct predictions. Class F, however, exhibits only modest accuracy. These results underscore the model's bias toward the majority class and indicate that class imbalance significantly affects its overall performance on less frequent arrhythmia types. [1]

The Decision Tree model achieved a training accuracy of 98.35% and a test accuracy of 95.65%. This performance indicates strong generalization capabilities with minimal overfitting. The slight drop in performance on the test set is expected and suggests that the model maintains robust predictive capabilities when presented with unseen data. [1]

5.2 Sophisticated Models

The sophisticated models generally demonstrated superior predictive power, with ensemble methods leading the performance metrics. XGBoost achieved the highest overall performance among all models evaluated, closely followed by LightGBM, indicating their efficacy in handling complex, high-dimensional data. Random Forest also achieved high precision, suggesting it makes fewer false positive predictions. In contrast, Linear SVC significantly underperformed, particularly in recall and F1-score, highlighting its inability to effectively generalize across all classes in this complex dataset. SVC also lagged behind the ensemble methods, further emphasizing the superiority of tree-based approaches for arrhythmia classification tasks. [1]

The overall performance metrics for these sophisticated models are presented in Table 4. [1]

Table 4: Performance Metrics for Sophisticated Models

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	0.9813	0.9599	0.8548	0.8996
LightGBM	0.9801	0.9455	0.8440	0.8872
Random Forest	0.9754	0.9702	0.8133	0.8766
SVC	0.9670	0.9436	0.7565	0.8266
Linear SVC	0.9041	0.8258	0.4511	0.4904

XGBoost Detailed Performance

XGBoost's performance across the five arrhythmia classes is detailed in Table 5. Class N, with the highest support of 14,577 instances, was classified with exceptional precision (0.9831) and recall (0.9969), resulting in an F1-score of 0.9900. This near-perfect performance on the dominant class significantly contributes to the model's high overall accuracy. However, Class S (Supraventricular premature beat) showed a much lower recall of 0.6818 despite a high precision of 0.9532, indicating a considerable number of false negatives, likely attributable to its low support (418 instances). Class V (Premature ventricular contraction) maintained a strong balance between precision (0.9560) and recall (0.9313), with an F1-score of 0.9435, reflecting stable model performance even with a smaller number of examples (1,120). Class F (Fusion of ventricular and normal beat) exhibited a similar issue to Class S, with a relatively low recall of 0.6842 and an F1-score of 0.7820, also attributable to its minority status (152 instances). Class Q (Unclassifiable beat) performed exceptionally well, with an F1-score of 0.9874. The macro average F1-score of 0.8996 demonstrates the model's overall class-wise performance, while the weighted average F1-score of 0.9803 indicates the model's high effectiveness driven primarily by its accuracy on high-support classes. [1]

Table 5: XGBoost Classification Metrics

Class	Precision	Recall	F1-Score	Support
0 (N)	0.9831	0.9969	0.9900	14,577
1 (S)	0.9532	0.6818	0.7950	418
2 (V)	0.9560	0.9313	0.9435	1,120
3 (F)	0.9123	0.6842	0.7820	152
4 (Q)	0.9951	0.9799	0.9874	1,244
Accuracy			**0.9813**	**17,511**
Macro Avg	0.9599	0.8548	0.8996	17,511
Weighted Avg	0.9809	0.9813	0.9803	17,511

The confusion matrix for XGBoost visually reinforces these findings. Class N is classified with near-perfect accuracy, with 14,532 of 14,577 samples correctly predicted, confirming the model's strong performance on the dominant class. However, Class S suffers from confusion with Class N, with 128 instances misclassified, significantly affecting its recall. Similarly, while Class V is mostly well classified, 69 instances are misclassified as Class N. Class F shows considerable confusion, particularly with Class N and Class V, reflecting the model's difficulty in distinguishing between certain abnormal rhythms. Overall, XGBoost handles the dominant class extremely well and performs decently on minority classes, though some class overlap remains a challenge, especially for rare arrhythmia types. [1]

XGBoost demonstrates near-perfect training accuracy (0.9990) and excellent test accuracy (0.9813). This minimal drop from training to test performance indicates strong generalization capabilities and highlights the model's ability to learn complex patterns with negligible overfitting. [1]

LightGBM Detailed Performance

LightGBM achieved a high test accuracy of 0.9801, closely rivaling XGBoost in overall performance. The model performed exceptionally well on Class N (F1-score: 0.9895) and Class Q (F1-score: 0.9874). However, similar to XGBoost, LightGBM struggled slightly with Class S and Class F, where recall dropped below 0.68, indicating potential class confusion for these minority classes. The macro average F1-score of 0.8872 demonstrates a strong overall balance in performance across classes, while the high weighted average F1-score of 0.9791 confirms LightGBM's effectiveness in handling imbalanced data distributions. [1]

The class-wise performance metrics for LightGBM are detailed in Table 6. [1]

Table 6: LightGBM Classification Metrics

Class	Precision	Recall	F1-Score	Support
0.0	0.9825	0.9966	0.9895	14577
1.0	0.9527	0.6746	0.7899	418
2.0	0.9513	0.9250	0.9380	1120
3.0	0.8448	0.6447	0.7313	152
4.0	0.9959	0.9791	0.9874	1244
Accuracy			**0.9801**	**17511**
Macro Avg	0.9455	0.8440	0.8872	17511

Weighted Avg	0.9795	0.9801	0.9791	17511

LightGBM achieved a high training accuracy of 0.9981 and a nearly equivalent test accuracy of 0.9801. This indicates excellent generalization capabilities with negligible overfitting. The marginal drop from training to test accuracy confirms the model's ability to maintain predictive reliability on unseen data, further reinforcing its strong performance. The confusion matrix for LightGBM also reveals strong performance on Class N (14,528 correct predictions), with some misclassifications occurring among minority classes, especially Class S, which is often confused with N. Class V is also accurately predicted, while classes F and Q exhibit a small number of errors. [1]

5.3 Deep-Learning Model - CNN

The Convolutional Neural Network (CNN) model achieved a strong overall accuracy of 0.84 across a balanced dataset, which was created through down-sampling for CNN training. The CNN performed best on classes 3 and 4, both reaching precision and recall values greater than or equal to 0.90, suggesting the model effectively captures distinct features of these arrhythmia types. Class 0, however, showed the weakest precision (0.69) but reasonable recall (0.84), indicating a propensity for false positives in this category. The macro and weighted averages for all metrics (precision, recall, F1-score) were consistently 0.84, demonstrating reliable and uniform classification performance across all classes within this downsampled and balanced dataset. [1]

The class-wise performance metrics for the CNN are detailed in Table 7. [1]

Table 7: CNN Classification Metrics

Class	Precision	Recall	F1-Score	Support
0	0.69	0.84	0.76	150

1	0.85	0.71	0.77	150
2	0.85	0.79	0.82	150
3	0.90	0.91	0.90	150
4	0.96	0.96	0.96	150
Accuracy			**0.84**	**750**
Macro Avg	0.85	0.84	0.84	750
Weighted Avg	0.85	0.84	0.84	750

The CNN shows a training accuracy of 0.92 and a lower test accuracy of 0.84, revealing some degree of overfitting. While the model performs well on the training data, the gap suggests it could benefit from additional regularization techniques or data augmentation to improve its generalization to unseen samples. The confusion matrix for the CNN model displays balanced performance across classes, with the highest precision seen in class 4 (195 correct) and class 2 (162 correct). However, there is visible confusion between class 0 and others, particularly class 1 (25 misclassified as 0). Despite this, each class retains high true positive counts, indicating that the model effectively captures distinguishing features across arrhythmia types. The CNN's training loss decreases steadily over the epochs, while training accuracy rises and plateaus above 0.90. This trajectory suggests effective learning with no indication of underfitting during the training phase itself. [1]

6. Limitations

While ensemble models like XGBoost deliver superior accuracy, their adoption comes with notable drawbacks, particularly concerning computational complexity. XGBoost, relying on an ensemble of decision trees, demands substantial memory and processing power. This characteristic renders it impractical for deployment on resource-constrained platforms such as wearable ECG monitors, where factors like battery life and low-latency response are paramount. In real-world scenarios, the high computational load associated with XGBoost could result in slow inference times and increased energy consumption, thereby undermining its efficacy for real-time performance in continuous monitoring applications. [1]

Similarly, the K-Nearest Neighbors (KNN) algorithm presents significant limitations for embedded applications. As a lazy learning model, KNN necessitates storing the entire training dataset. During inference, it performs distance calculations for each new input against all stored training examples. This leads to high memory usage and considerable latency challenges that are particularly detrimental for continuous ECG monitoring on wearable devices. The slow response time inherent in KNN can impair timely clinical decision-making, making it unsuitable for real-time, on-device analysis where immediate feedback is often critical. [1]

7. Future Scope

The future trajectory of this project is directed towards enabling the practical deployment of arrhythmia detection models on wearable devices and ensuring their real-world effectiveness. This involves several key areas of development. [1]

A primary direction for future work is the **optimization of models for wearable devices**. This is crucial given the inherent limitations of wearable platforms in terms of battery life and processing power, which necessitate low-latency responses. To achieve this, techniques such as pruning (removing unnecessary connections or neurons from the network), quantization (reducing the precision of numerical representations of weights and activations), and the utilization of TinyML frameworks are essential. Frameworks like TensorFlow Lite Micro and Edge Impulse offer practical toolsets for reducing model size, ideally to under 100 KB, which would enable deployment on low-power microcontrollers commonly found in devices like the Apple Watch. [1]

Additionally, **integrating edge computing** represents a significant future step. This approach can help offload complex computations to the cloud while simultaneously maintaining real-time responsiveness directly on the device. This hybrid model allows for more intensive processing to occur remotely, reducing the burden on the wearable device itself. [1]

Designing a **robust, real-time processing pipeline** is also crucial. This comprehensive pipeline would encompass every stage from ECG signal acquisition from the wearable device and efficient on-device pre-processing to performing inference with the optimized machine learning model and generating timely alerts for potential cardiac abnormalities. Such a pipeline is vital to ensure seamless and effective continuous monitoring. [1]

Finally, **clinical validation** remains an indispensable step. This involves establishing collaborations with medical institutions and conducting trials in real-world environments. The objective of these trials is to rigorously assess the model's reliability and effectiveness in diverse patient populations and conditions. A critical aspect of this validation is to minimize false positives, which can lead to unnecessary anxiety and medical interventions, while simultaneously guaranteeing the effective detection of life-threatening arrhythmias in everyday use, thereby providing genuine clinical value. [1]

8. Conclusion

The comprehensive evaluation of various machine learning models for arrhythmia classification underscores a critical balance between predictive performance and practical deployment feasibility. While complex models often yield top-tier accuracy, their computational demands can present significant barriers for real-world applications, particularly on resource-constrained platforms. [1]

Among the sophisticated models, LightGBM emerges as a strong candidate. It offers high accuracy, with an F1-score of approximately 0.8872, while demonstrating greater resource efficiency compared to XGBoost. This efficiency stems from its optimized tree growth and sampling methods, making it particularly suitable for scenarios with moderate hardware constraints where a balance between performance and computational overhead is required. [1]

Conversely, Decision Trees, categorized as simpler models, achieve an impressive F1-score of around 0.9546 and are highly efficient in both memory usage and inference time. Their lightweight nature makes them an ideal choice for deployment on low-power devices, where computational resources are severely limited. Moreover, the inherent interpretability of Decision Trees is a key strength, especially in clinical settings. In healthcare, transparency in the decision-making process is crucial for fostering trust among medical professionals and enabling informed interventions, a factor where complex "black box" models often fall short. [1]

These findings collectively suggest that while complex models like XGBoost and LightGBM provide superior predictive power, simpler, more interpretable models such as Decision Trees may offer a more practical and trustworthy solution for real-world, resource-constrained applications. For continuous health monitoring via wearable devices, the ability to provide accurate, timely, and understandable diagnostic information, even with a slight trade-off in peak accuracy, can be more valuable than raw predictive power alone. The choice of model, therefore, depends heavily on the specific deployment environment and the relative importance of interpretability versus raw predictive accuracy. [1]

9. Code and Data Access

To access the project's Python file and view the MIT-BIH train and test datasets, please scan the QR code provided in the original document's Appendix. This QR code grants access to the code and datasets. [1]

10. Bibliography

Z. S. M. J. L. M. P. R... Mar T, Optimization of ECG classification by means of feature selection, « IEEE. [1]

H. L. A. N. M. F. Daamouche A, »A wavelet optimization approach for ECG signal classification.«. 1

K. M. H. A. Houssein EH, »ECG signals classification: a review« 1

F. S. S. M. Kachuee M, »Ecg heartbeat classification: A deep transferable representation.«. 1

W. P., »Support vector machine learning for ECG classification.«. 1

L. M. D. M. G. A. Ebrahimi Z, A review on deep learning methods for ECG arrhythmia classification «. 1

C. T. Weimann K, Transfer learning for ECG classification«. 1

Works cited

1. accessed January 1, 1970,