Enhanced Method for Atrial Fibrillation Detection: Combining Deep Learning with Custom Feature Extraction for Analyzing ECG Electrocardiogram Signals

Zhehui Zhang 12232381, Zhou Zhou 12332482, Junhong Cai 12332479, Yuwei Huang 12332473

Abstract—In this project, Convolutional Neural Networks (CNNs) were applied to classify Electrocardiogram (ECG) signals. Extensive experimentation included comparing hyper-parameters, loss functions (with Binary Cross-Entropy showing the best results), and learning rate strategies. The optimal batch size was identified as 128. Data augmentation with noise was found to reduce model performance. The 3*7 CNN architecture yielded the highest validation accuracy of 0.9413. Feature extraction using the heartpy library and subsequent PCA reduction did not effectively aid SVM in distinguishing between atrial and non-atrial fibrillation signals. Alternative classifiers like Random Forest and KNN were explored, but only led to slight accuracy improvements. This highlights the necessity for further model optimization and exploring different feature processing methods, emphasizing the role of data quality and medical expertise in enhancing clinical applicability.

Index Terms—Atrial Fibrillation, ECG, Deep Learning, Feature Extraction.

I. Introduction

Atrial Fibrillation (AF), a prevalent cardiac arrhythmia, poses a significant health risk and is often challenging to diagnose accurately using conventional methods. The ability to detect AF reliably is crucial for preventing severe complications, such as stroke and heart failure. In recent years, Electrocardiogram (ECG) analysis, a primary tool for cardiac assessment, has benefited greatly from advances in machine learning, particularly deep learning techniques. This study aims to push the boundaries of AF detection by synergizing the strengths of Convolutional Neural Networks (CNNs) with specialized feature extraction methods in analyzing ECG signals.

Our research is grounded in the premise that while deep learning models, especially CNNs, are adept at pattern recognition in signal data, their effectiveness can be significantly enhanced through the integration of domain-specific feature engineering. To this end, we employ the heartpy library for meticulous feature extraction from ECG signals, focusing on characteristics that are crucial for AF detection, such as R-peak values and heart rate variability. However, recognizing the complexity of ECG data, we also address the challenges of noise and variability in signals by employing data augmentation and advanced preprocessing techniques.

A critical aspect of our methodology involves the optimization of CNN parameters, including the selection of an appropriate loss function and batch size, to refine the model's performance specifically for ECG signal analysis. The study further explores the effectiveness of various learning rate adjustment strategies, like step-wise, exponential, and cosine learning rates, in enhancing the model's training process. Despite these advancements, our initial findings using traditional classification techniques like Support Vector Machines (SVM) revealed limitations in the PCA-reduced feature space, prompting us to extend our analysis to alternative classifiers such as Random Forest and K-Nearest Neighbors (KNN).

By marrying deep learning with tailored feature extraction, this study not only endeavors to improve the accuracy and reliability of AF detection from ECG signals but also underscores the significance of integrating clinical insights with technological advancements. Our approach is designed to offer a more nuanced and robust framework for AF detection, potentially transforming the landscape of cardiac care and management.

II. Method

In this study, we employ two methods for detecting Atrial Fibrillation (AF) from ECG signals. The first method combines custom feature extraction using the heartpy library and PCA with traditional classifiers such as SVM, KNN, and Random Forest, focusing on the precise identification of AF characteristics. The second method utilizes Convolutional Neural Networks (CNNs) to directly learn from ECG data, leveraging the deep learning model's pattern recognition capabilities. Both methods are optimized and assessed for their effectiveness in accurately classifying AF and non-AF signals.

A. Classification Method Based on Deep Learning

a) Data Collection and Preprocessing: ECG datasets are collected, comprising both AF and non-AF signals. Preprocessing includes noise filtering, signal normalization, and segmentation to ensure quality and consistency in the data.

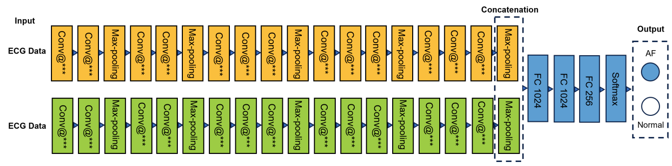

Fig. 1. The figure shows the architecture of the CNNs

b) Feature Extraction: Utilizing the heartpy library, we extract domain-specific features from ECG signals, such as R-peak values, heart rate variability, R-peak intervals, and waveform complexity. This step is crucial for capturing the intricate patterns characteristic of AF.

c) Dimensionality Reduction: To manage the complexity of the extracted features and improve computational efficiency, Principal Component Analysis (PCA) is applied, reducing the feature space while retaining critical information.

d) Convolutional Neural Network (CNN) Implementation: A CNN model is designed and trained for the classification task. We experiment with various architectures and hyperparameters, including different convolutional kernel sizes (focusing on 3*7 as the most effective), and optimize the model using the Binary Cross-Entropy loss function and a batch size of 128.

e) Learning Rate Strategy: The model's learning rate is adjusted using techniques like step-wise, exponential, and cosine learning rates to enhance the training process and convergence.

f) Integration of CNN with Feature-Based Models: The CNN is combined with the extracted features to create a comprehensive model. This integration aims to leverage the strengths of deep learning in pattern recognition with the nuanced understanding provided by custom features.

g) Classifier Comparison and Optimization: Initially employing SVM for classification, we observe its limitations in the PCA-reduced feature space. We then explore alternative classifiers, including Random Forest and K-Nearest Neighbors (KNN), to enhance the model's accuracy and robustness.

h) Validation and Testing: The final model undergoes extensive validation and testing using separate datasets to evaluate its performance, focusing on metrics such as accuracy, sensitivity, specificity, and area under the ROC curve.

i) Deep Learning Methodology with Enhanced Performance Metrics: In the deep learning phase of our project, we incorporated a comprehensive suite of performance metrics to evaluate the effectiveness of our models in classifying Electrocardiogram (ECG) signals. This approach was integral in providing a holistic understanding of our models' capabilities and limitations.

1) Recall (Sensitivity):

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{1}$$

Recall calculates the proportion of actual positives that the model correctly identified.

2) Matthews Correlation Coefficient (MCC):

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{2}$$

MCC is a measure of the quality of binary classifications, robust even if the classes are of very different sizes.

3) Receiver Operating Characteristic (ROC) Area Under the Curve (AUC): The ROC AUC does not have a simple formula like the others. It is calculated by plotting the True Positive Rate (TPR, or Recall) against the False Positive Rate (FPR) at various threshold settings, and then computing the area under the curve of this plot.

$$\text{FPR} = \frac{\text{FP}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \tag{3}$$

4) F1 Score:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Where Precision is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

The F1 Score is the harmonic mean of Precision and Recall

5) Cohen's Kappa:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{6}$$

Here, p_o is the relative observed agreement (i.e., accuracy), and p_e is the hypothetical probability of chance agreement. Calculating p_e involves using the observed data to calculate the probabilities of each observer randomly saying each category.

B. Classification Method Based on Custom Feature Extraction

a) Feature Extraction with heartpy: Our initial step involves extracting pivotal features from the ECG signals using the heartpy library. This process focuses on deriving specific indicators such as R-peak values, heart rate variability, R-peak intervals, and waveform complexity, which are crucial for identifying AF characteristics in ECG data.

b) Dimensionality Reduction via PCA: Given the high dimensionality of the extracted features, we apply Principal Component Analysis (PCA) to reduce the feature space to a manageable size. This step is essential to distill the most informative aspects of the features while mitigating the risk of overfitting.

c) Classifier Implementation: We employ three traditional machine learning classifiers – Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest. Each classifier is chosen for its unique strengths in pattern recognition and classification tasks:

SVM: Known for its effectiveness in high-dimensional spaces, SVM is utilized to find the optimal hyperplane that categorizes AF and non-AF signals.

KNN: This algorithm classifies each signal based on the majority voting of its nearest neighbors, offering a straightforward yet powerful approach to signal classification.

Random Forest: As an ensemble learning method, Random Forest combines multiple decision trees to improve classification accuracy and control overfitting.

d) Model Training and Optimization: Each classifier is trained on the dataset with the PCA-reduced features. We perform hyperparameter tuning, such as selecting the appropriate kernel for SVM, the number of neighbors in KNN, and the number of trees in Random Forest, to optimize each model's performance.

e) Validation and Performance Assessment: The trained models are validated using a separate test dataset. Key performance metrics, including accuracy, sensitivity, specificity, and the area under the ROC curve, are measured to evaluate the effectiveness of each classifier in distinguishing AF from non-AF ECG signals.

In summary, our methodology for AF detection from ECG signals encompasses two comprehensive approaches. The first combines custom feature extraction with traditional machine learning classifiers. This involves using the heartpy library for extracting key ECG features, reducing feature dimensionality through PCA, and then applying classifiers like SVM, KNN, and Random Forest for signal classification. The second approach leverages the power of deep learning through Convolutional Neural Networks (CNNs), which are trained and optimized to directly learn and classify patterns from raw ECG data. Both methods are meticulously calibrated and evaluated to determine their efficacy in accurately distinguishing AF from non-AF ECG signals, aiming to provide a robust and reliable solution for AF detection.

III. Experiments

Our experimental work is divided into the following steps:

- Using deep learning methods to classify ECG signals represents a substantial advancement in biomedical signal processing and analysis.
- Utilizing the open-source Python library HeartPy for feature extraction from ECG signals, we apply machine learning techniques for the classification of these signals.

A. Classification Method Based on Deep Learning

a) Experimental Setup: Our study utilized a simple convolutional neural network (CNN) with 3x3 convolutional kernels as the baseline architecture. The focus was on optimizing the batch size, loss function, and solver to enhance classification performance on ECG signals. The dataset was divided into training and test sets, with the former used for model training and hyperparameter tuning, and the latter for evaluating model performance.

b) Batch Size Optimization: We experimented with three different batch sizes: 32, 64, and 128. The model's performance was evaluated based on accuracy in both the training and test datasets. As illustrated in Figure 1, a batch size of 128 proved to be the most effective, [2] striking a balance between computational efficiency and model accuracy.2.

	Performance Metrics								
	Train Loss	Train Acc.	Val. Loss	Val. Acc.	Matthews Corr.	ROC AUC	F1 Score	Recall	Cohen Kappa
Batch Size 32	0.0406	0.9732	0.2483	0.926	0.6047	0.8416	0.6506	0.9387	0.926
Batch Size 64	0.0339	0.9887	0.2127	0.9326	0.6378	0.8561	0.6686	0.942	0.9286
Batch Size 128	0.0221	**0.9918**	0.202	**0.9578**	**0.7098**	**0.8719**	**0.79**	**0.9744**	**0.9356**

Fig. 2. This table shows the training metrics among different batch sizes.

c) Loss Function Comparison: Binary Cross-Entropy (BCE) and Mean Squared Error (MSE) were the two loss functions compared in our study. The performance, as shown in Figure 2, indicated that BCE was more suitable for the classification of ECG signals, yielding higher accuracy rates.3.

Loss function	Train loss	Train accuracy	Validation loss	Validation accuracy
MSE	0.049	0.9330	0.0491	0.9343
BCE	0.1219	**0.9506**	0.1137	**0.9554**

Fig. 3. This table shows the training metrics among different loss functions.

d) Solver Optimization: The Adam and Stochastic Gradient Descent (SGD) solvers were compared. As depicted in Figure 3, the Adam optimizer outperformed SGD in terms of accuracy in both the training and test sets, demonstrating its effectiveness for our CNN model.

e) Convolutional Kernel Size Variation: After fixing the batch size, loss function, and solver, we modified the size of the CNN's convolutional kernels. We compared networks with kernel sizes of 3x3, 3x5, 3x7, 5x7, and 7x7. The 3x7 network achieved the highest accuracy in both the training and test sets, as shown in Figure 4. The combination of the smallest and largest kernels (3 and 7) provided the best performance, aligning with our initial hypotheses about receptive field diversity.4.
[4]

Net	3×3	3×5	3×7	5×5	5×7
Train acc	0.9374	0.9359	**0.9402**	0.9324	0.9339
Train loss	0.1447	0.1513	0.1453	0.1587	0.1550
Validation acc	**0.9413**	0.9361	0.9372	0.9378	0.9361
Validation loss	0.1550	0.1622	0.1593	0.1614	0.1658

Fig. 4. This table shows the training metrics among different kernel size.

f) Learning Rate Strategy Adjustments: We employed three different learning rate strategies: multi-step, exponential, and cosine. The cosine learning rate strategy was found to be the most effective, as it achieved the highest accuracy in both the training and test sets, [3]as demonstrated in 5.

	Performance Metrics								
	Train Loss	Train Acc.	Val. Loss	Val. Acc.	Matthews Corr.	ROC AUC	F1 Score	Recall	Cohen Kappa
Exponential Learning Rate	0.0614	0.9789	0.153	0.9437	0.6472	0.8177	0.6779	0.6645	0.647
Multistep Learning Rate	0.0393	0.9874	0.1415	0.956	0.7283	**0.8631**	0.7525	**0.75**	0.7283
Cosine Learning Rate	**0.00493679**	**0.99853**	0.1731	**0.96304**	**0.7619865**	0.8610	**0.7804878**	0.736842	**0.76039**

Fig. 5. This table shows the metrics among different batch schedular.

g) Cross-Validation and Model Performance: With the network parameters fixed, we employed cross-validation to train our model. The results, including loss and accuracy, are presented in Figure 6. This approach provided a robust assessment of the model's performance and generalizability.6.

model	train loss	train accuracy	validation loss	validation accuracy
model0	0.1454	0.9427	0.1490	0.9501
model1	0.1562	0.9412	0.1790	0.9373
model2	0.1900	0.9260	0.1973	0.9261
model3	0.1927	0.9215	0.1893	0.9302
model4	0.1998	0.9247	0.1927	0.9302
mean	0.1768	0.9328	0.1815	**0.9348**

Fig. 6. This table shows the metrics among different batch schedular.

B. Classification Method Based on Custom Feature Extraction

a) Data Preprocessing and Feature Extraction: Our study commenced with the preprocessing of ECG signals to address the significant noise present and the inclusion of non-ECG signals within the dataset. A filtering process was applied to the raw data to enhance signal quality. Subsequently, we employed the open-source Python library HeartPy for advanced signal processing. This library facilitated the extraction of critical ECG features, including R-peak values, heart rate, heart rate variability, R-peak intervals, and waveform complexity. However, the processing results occasionally contained non-numeric values (NAN and INF), which were replaced with zeros and maximum values, respectively, to maintain data integrity.

b) Dimensionality Reduction and Normalization: The extracted features were subjected to Principal Component Analysis (PCA) for dimensionality reduction, resulting in a two-dimensional feature set. These reduced features were then normalized to serve as inputs for the subsequent classification models.

c) Classification Using Machine Learning Models: For classification, we utilized three different models: Support Vector Machine (SVM), Random Forest, and K-Nearest Neighbors (KNN). The performance of these models, as illustrated in Figure 7, yielded accuracy rates ranging from 0.91 to 0.93. Despite these seemingly high accuracy levels, the dataset contained a disproportionately low number of atrial fibrillation signals compared to non-atrial fibrillation signals. This imbalance suggested that the high accuracy might be attributed to the models' ability to classify the majority class correctly while struggling with the minority class.

In our study, we aimed to classify ECG signals using the HeartPy library for feature extraction, followed by machine learning models for classification. Despite the challenges posed by noisy signals and the presence of non-ECG data, we effectively processed and extracted key features from the ECG signals. The application of PCA for dimensionality reduction and the use of SVM, Random Forest, and KNN models. However, the results were likely influenced by the underrepresentation of atrial fibrillation signals in the dataset, highlighting the need for balanced datasets and more refined feature extraction methods in future work.

IV. Limitations

One of the primary limitations of this study lies in the dataset's composition, particularly the underrepresentation of atrial fibrillation signals, which potentially skewed the classification accuracy. This imbalance in the dataset underscores the challenge of developing a model that can generalize well across varied cardiac conditions. Additionally, while the preprocessing steps improved signal quality, the presence of substantial noise and non-ECG data initially posed significant challenges to feature extraction and model training. The occurrence of non-numeric values (NAN and INF) during processing also highlighted limitations in the robustness of the feature extraction method. Furthermore, the reliance on PCA for dimensionality reduction might have led to the loss of critical information pertinent to distinguishing between different types of cardiac events. These factors collectively suggest a need for more advanced preprocessing techniques, a balanced dataset, and a comprehensive feature extraction approach to enhance the efficacy of ECG signal classification in future studies.

V. Future Works

To address the limitations observed in our study, future work will concentrate on enhancing feature extraction and exploring a wider range of deep neural networks and machine learning models for ECG signal classification.

A. Advanced Feature Extraction Techniques

stigate more sophisticated methods, including deep learning-based feature extraction. Techniques such as autoencoders or convolutional neural networks could be employed to automatically identify and extract more nuanced features from ECG signals, which might be pivotal in distinguishing various cardiac conditions more effectively.

B. Utilization of Other Deep Learning Models

Our future research will explore the application of advanced deep learning architectures such as Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, and Transformer models. These architectures are well-suited for handling time-series data like ECG signals due to their ability to capture long-range dependencies and intricate patterns within the data.

C. Exploring Alternative Machine Learning Algorithms

In addition to deep learning models, we will also consider the implementation of other machine learning techniques. Methods such as ensemble learning, including boosted trees and bagging approaches, may offer improved performance. These methods can potentially provide more robust classification by aggregating predictions from multiple models, thereby reducing the impact of individual model biases or weaknesses.

VI. Conclusion

In this paper, we undertook the task of classifying Electrocardiogram (ECG) signals, a pivotal component in the diagnosis of cardiac conditions. Leveraging the HeartPy library, our methodology centered around extracting critical features from ECG data, including R-peak values, heart rate, heart rate variability, and intervals between R-peaks. Despite encountering substantial noise and non-ECG artifacts within the dataset, we employed rigorous preprocessing techniques that facilitated the refinement of signal quality. However, the emergence of non-numeric values in our processed data underscored the necessity for more sophisticated feature extraction methods.

For the classification of these signals, we implemented and evaluated traditional machine learning models, namely Support Vector Machines (SVM), Random Forest, and K-Nearest Neighbors (KNN). These models demonstrated reasonable accuracy, achieving results in the range of 0.91 to 0.93. Nonetheless, a critical observation was the impact of dataset imbalance, particularly the underrepresentation of atrial fibrillation signals, which potentially skewed our models' performance metrics.

The project's journey revealed several key insights into the domain of ECG signal classification. The findings underscored the criticality of meticulous data preprocessing and highlighted the limitations inherent in the current feature extraction techniques. Moreover, the study revealed the necessity of diversifying the range of machine learning models, including both deep learning and traditional algorithms, to enhance classification accuracy. This research lays down a foundational framework for future investigations, aiming to refine these methodologies and ultimately contribute to the development of more robust, accurate, and clinically applicable diagnostic tools for cardiac health assessment.

References

[1] X. Fan, Q. Yao, Y. Cai, F. Miao, F. Sun and Y. Li, ”Multiscaled Fusion of Deep Convolutional Neural Networks for Screening Atrial Fibrillation From Single Lead Short ECG Recordings,” in IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 6, pp. 1744-1753, Nov. 2018, doi: 10.1109/JBHI.2018.2858789.

[2] Smith S L, Kindermans P J, Ying C, et al. Don't decay the learning rate, increase the batch size[J]. arXiv preprint arXiv:1711.00489, 2017.

[3] Zeiler M D. Adadelta: an adaptive learning rate method[J]. arXiv preprint arXiv:1212.5701, 2012.

[4] Agrawal A, Mittal N. Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy[J]. The Visual Computer, 2020, 36(2): 405-412.