Project Report

Project Title: Comparing ECG-based classification with CNN, Decision Trees, and KNN

Name: Geoffrey Cloud

Date: May 4th, 2025

Course: Introduction to Machine Learning

Literature Review

Electrocardiogram (ECG) signals are vital for diagnosing and monitoring cardiovascular health, offering detailed insight into the electrical activity of the heart. In recent years, machine learning and deep learning models have shown promise in automating ECG signal classification for detecting abnormal patterns. This project aims to compare the performance of different classification models, specifically Convolutional Neural Networks (CNN), K-Nearest Neighbors (KNN), and Decision Trees when classifying ECG data. By evaluating these models on the same dataset, the study highlights their relative strengths, weaknesses for real world medical applications.

Method A - Convolutional Neural Networks:

- Authors: Mohamad M. Al Rahhal, Yakoub Bazi, Mansour Al Zuair, Esam Othman, Bilel BenJdira
- Title: *Convolutional Neural Networks for Electrocardiogram Classification*
- Year: 2018
- Journal: Journal of Medical and Biological Engineering
- Link: Convolutional Neural Networks for Electrocardiogram Classification

In the study *Convolutional Neural Networks for Electrocardiogram Classification,* The authors proposed a deep learning approach for ECG-based signal classification by transforming one-dimensional ECG signals into two dimensional image representations using the Continuous Wavelet Transform (CWT). This transformation enables the use of pre-trained Convolutional Neural Networks (CNNs), such as VGGNet, which require red, green, and blue (RGB) image inputs. By converting ECG signals into time and frequency domain representations, the authors were able to leverage powerful feature extraction capabilities that CNNs have to offer. The RGB images that were generated via CWT were input into the CNN to extract high level features, which were then fed into a fully connected neural network for classification. To help reduce overfitting during the training process, the dropout regularization was applied. This method demonstrated the effectiveness of combining signal transformation with deep learning architectures for improved accuracy in ECG-based signal classification.

Method B - Decision Trees:

- Author: Runqiang Xing
- Title: *Classification and Prediction of ECG Data Based on Decision Trees and their Optimization Algorithms*
- Year: 2023
- Conference: 2023 International Conference on Computers, Information Processing and Advanced Education (CIPAE)
- Link: Classification and Prediction of ECG Data Based on Decision Trees and their Optimization Algorithms

The author of this study proposed a method for classifying cardiac arrhythmias from ECG data using a decision tree based machine learning model. The model was trained on key statistical features extracted from ECG signal segments including mean, variance, maximum and sum. Initially, the model was used to classify ECG signal segments into six distinct risk levels ranging from low to high risk. While the model achieved a low accuracy score of roughly 39%, insights from the false alarm analysis highlighted the model's general effectiveness in distinguishing between risk levels.

In order to optimize the model and improve its sensitivity to dangerous conditions, the author simplified the classification task by binarizing classes into two groups, high risk and low risk, so that it will enhance sensitivity when detecting critical cases. This study demonstrates the potential for using lightweight and interpretable decision tree models for real time ECG monitoring which can be integrated into ECG systems for early detection in a timely manner.

Algorithm Implementation

Method A:

Before training the Convolution Neural Network, the ECG signals were preprocessed using the *scipy.signal.cwt* function with the Ricker wavelet in order to extract time and frequency features. After transformation, the resulting images were normalized and reshaped to include an RGB channel with 3 channels per image, which is needed for compatibility for CNN input. Next, the corresponding labels were preprocessed using one-hot encoding to support multi-class classification.

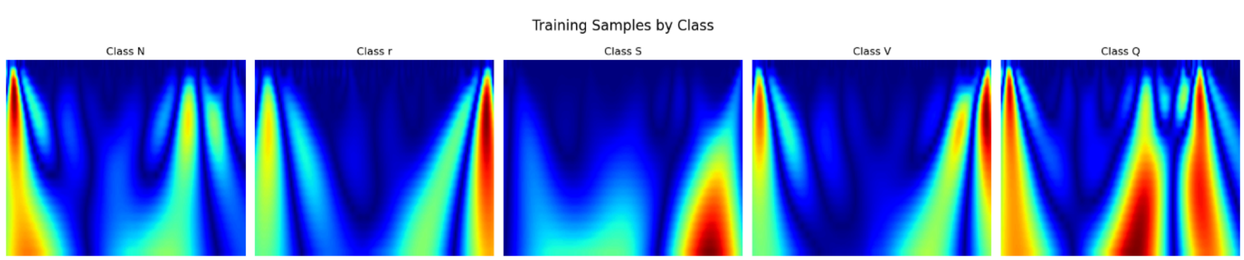

Figure 1. Preprocessed Data Samples using Continuous Wave Transform. [ECG-classification-CNN.ipynb]

The model was built using TensorFlow and consists of a four layer deep neural network. The first two layers are convolutional layers that extract features from the CWT transformed images with 128, and 256 filters respectively, with both having ReLU activation functions. Each of the layers are followed by pooling in order to reduce dimensionality for the neural network classifier. The final two layers (neural network classifier) consist of a dense layer with 128 neurons using ReLU as an activation function and an output layer with 5 neurons corresponding to each ECG class using Softmax as an activation function .

Method B:

The decision tree classifier was implemented to categorize ECG signals through extracted statistical features. The preprocessing phase involved applying a custom 'ExtractStatistics' transformer within a pipeline to compute the statistics mentioned in this method proposed by the author: mean, variance, sum, and maximum. These features were selected to reduce the dimensionality of the data while also preserving important characteristics.

The model itself was configured with a maximum depth of 6, a minimum of 10 samples required to split a node, and a minimum of 5 samples required to be at a leaf node. These hyperparameters are suggested in the article as the most optimal in order to prevent overfitting.

Experiments

Method A (CNN) vs Method B (Decision Tree)

-----------Convolutional Neural Network Classification Report-------

	precision	recall	f1-score	support
N	0.97	0.99	0.98	781
r	0.94	0.86	0.90	590
S	0.57	0.53	0.55	43
V	0.37	0.55	0.44	75
Q	0.27	0.27	0.27	11
accuracy			0.90	1500
macro avg	0.63	0.64	0.63	1500
weighted avg	0.91	0.90	0.91	1500

--------------Decision Tree classification report-----------------

	precision	recall	f1-score	support
N	0.89	0.99	0.94	781
r	0.92	0.88	0.90	590
S	0.67	0.09	0.16	43
V	0.53	0.28	0.37	75
Q	0.00	0.00	0.00	11
accuracy			0.88	1500
macro avg	0.60	0.45	0.47	1500
weighted avg	0.87	0.88	0.86	1500

The classification reports show that the Convolutional Neural Network outperformed the Decision Tree model, especially for the minority classes. Both models did well classifying the two majority classes N and r, with high precision and recall. However, the Decision Tree struggled to identify the minority classes S, V, and Q or missing them entirely (class Q). Overall the CNN achieved a higher accuracy of 90% vs 88%.

I believe the Decision Tree struggled with the minority classes likely due to overfitting to the majority classes. This model tends to perform well when there are enough samples in each class but fails to generalize properly when classes are extremely imbalanced.

Method A (CNN) vs Method C (KNN):

-----------Convolutional Neural Network Classification Report-------

	precision	recall	f1-score	support
N	0.97	0.99	0.98	781
r	0.94	0.86	0.90	590
S	0.57	0.53	0.55	43
V	0.37	0.55	0.44	75
Q	0.27	0.27	0.27	11
accuracy			0.90	1500
macro avg	0.63	0.64	0.63	1500
weighted avg	0.91	0.90	0.91	1500

--------------K-Nearest Neighbors classification report------------

	precision	recall	f1-score	support
N	0.97	0.99	0.98	781
r	0.92	0.86	0.89	590
S	0.62	0.35	0.45	43
V	0.33	0.48	0.39	75
Q	0.33	0.18	0.24	11
accuracy			0.89	1500
macro avg	0.63	0.57	0.59	1500
weighted avg	0.90	0.89	0.89	1500

Method A outperformed Method C in overall accuracy (90% vs 89%) and demonstrated better balance in precision, recall, and F1-score, especially for underrepresented classes. While method C achieved a high precision for the dominant class ‘N’ (F1: 0.98), its performance dropped significantly for minority classes ‘V’ and ‘Q’ (F1 scores: 0.39, and 0.23) respectively. In contrast, Method A showed improved recall on these minority classes, especially for class ‘V’, resulting in better overall class balance. Although both models showed strong performance on the dominant class, method A’s ability to learn more patterns gave it an edge in handling the class imbalance on the ECG dataset.

I believe method A outperformed method C because the extracted features from the convolutional layers in the CNN provides the neural network classifier intricate details that the KNN model just couldn’t see and learn from. The convolutional layers are designed to detect complex patterns and extract them.

Method B (Decision Tree) vs Method C (KNN)

-------------Decision Tree classification report----------------

	precision	recall	f1-score	support
N	0.89	0.99	0.94	781
r	0.92	0.88	0.90	590
S	0.67	0.09	0.16	43
V	0.53	0.28	0.37	75
Q	0.00	0.00	0.00	11
accuracy			0.88	1500
macro avg	0.60	0.45	0.47	1500
weighted avg	0.87	0.88	0.86	1500

-------------K-Nearest Neighbors classification report------------

	precision	recall	f1-score	support
N	0.97	0.99	0.98	781
r	0.92	0.86	0.89	590
S	0.62	0.35	0.45	43
V	0.33	0.48	0.39	75
Q	0.33	0.18	0.24	11
accuracy			0.89	1500
macro avg	0.63	0.57	0.59	1500
weighted avg	0.90	0.89	0.89	1500

Both the Decision Tree and KNN models show high performance on the majority classes N and r but both struggle with the minority classes S, V, and Q. Despite the poor performance, KNN performs slightly better in generalizing to the minority classes. Notably, KNN managed to make a few correct classifications for class Q while the Decision Tree completely misclassified all Q samples. Both models suffer from the same issue of class imbalance, which severely impacts their ability to classify the minority classes accurately.

Conclusions

For Method A, I don't believe that I was able to fully reproduce the results reported in the original paper's experiments. A potential reason for this is the choice of wavelet used during the Continuous Wavelet Transform (CWT) preprocessing step. The paper suggests using wavelets such as Daubechies (db4), Coiflets (coif3), and Biorthogonal (bior3.5). However, because of SciPy's cwt method, I used ricker as this is the default wavelet, although it probably is not the best wavelet for ECG signals. As a result, the resulting images from the CWT stage may not have been as detailed, which could be a reason for not performing as accurately.

Experiments with Decision Tree and K-Nearest Neighbors revealed a consistent pattern. It's clear that the models are biased toward majority classes. This finding highlights a key challenge with class imbalance. The Decision tree was more prone to overfitting on the dominant patterns within the data while KNN's reliance on local neighbors gave it a slight edge when classifying minority classes. I believe a few potential solutions for the issue would be to apply class rebalancing techniques such as imblearn's SMOTE or ADASYN. These techniques produce synthetic data and are meant for giving minority classes more representation during training. Another technique to address low performance for these models would be model tuning and threshold optimization.

In conclusion, while traditional models provide a good baseline, overcoming class imbalance and improving feature representation are key to achieving reliable ECG classification.

Sharing Agreement

Do you agree to share your work as an example for next semester? Yes

Do you want to hide your name/team if you agree? Yes