Classifying Heart Abnormalities Using ECG Signals and Machine Learning

Alaa Aldybous
College of Engineering,
Effat University
Jeddah, Saudi Arabia
Alnaldybous@effat.edu.sa

Mawa Hijji
College of Engineering,
Effat University
Jeddah, Saudi Arabia
Maahijji@effat.edu.sa

Enfel Barkat
College of Engineering,
Effat University, Saudi Arabia
ebarkat@effatuniversity.edu.sa

Liza Yousef
College of Engineering,
Effat University
Jeddah, Saudi Arabia
Lmyousef@effat.edu.sa

Jazzal Kandiel
College of Engineering,
Effat University
Jeddah, Saudi Arabia
Jankandiel@effat.edu.sa

Abstract - The electrocardiogram (ECG) is an essential diagnostic tool in cardiology, offering non-invasive insights into the heart's electrical activity. However, manual ECG interpretation is time-intensive and prone to errors, necessitating automated solutions. Machine learning (ML), particularly Convolutional Neural Networks (CNNs), has proven to be the most effective method for ECG classification, demonstrating superior accuracy and scalability. This paper reviews advancements in ML models, emphasizing CNNs and their methodologies[1][2]. It discusses preprocessing techniques[3], dataset requirements[4][5], challenges like imbalance and variability[6][7], and opportunities for real-time diagnostics. By addressing these factors, this paper highlights the transformative potential of CNNs in cardiovascular care and outlines future directions to enhance their clinical integration.

Keywords— Electrocardiograms, Premature Ventricular Contraction, Atrial Fibrillation, Ventricular Tachycardia

I. Introduction

Cardiovascular diseases (CVDs) remain the leading cause of mortality worldwide, responsible for nearly 18 million deaths annually. Electrocardiograms (ECGs) are indispensable for diagnosing CVDs, providing insights into arrhythmias such as Premature Ventricular Contraction (PVC), Atrial Fibrillation (AFib), and Ventricular Tachycardia (VTach)[3]. However, traditional manual ECG interpretation is resource-intensive, prone to variability, and often limited in its scalability [6].

Machine learning (ML), particularly Convolutional Neural Networks (CNNs), has revolutionized ECG analysis by automating feature extraction and delivering unparalleled accuracy. CNNs are highly effective for processing high-dimensional time-series data, making them well-suited for clinical and real-time applications [4][8]. This paper reviews CNN methodologies, highlights data preprocessing techniques, discusses challenges like dataset imbalance, and explores the implications of CNN-based systems for cardiovascular healthcare.

II. Literature Review

In recent years, using machine learning to classify ECG signals has emerged, enabling precise detection of heart abnormalities. Various machine learning models have been applied to classify ECG, each offering their advantages and their limitations. This literature review explores these different machine learning models comparing their accuracy and challenges.

A. Support Vector Machines (SVM)

SVM has been widely used for binary classification tasks, achieving up to 99.92% accuracy with metaheuristic optimization [7]. However, its limited scalability makes it less suitable for large-scale ECG datasets.

B. Artificial Neural Networks (ANN)

ANNs have demonstrated accuracies ranging from 90% to 98% in classifying arrhythmias [8]. However, they rely heavily on feature engineering, which restricts their adaptability compared to CNNs [6].

C. Long Short-Term Memory (LSTM) Networks

LSTMs excel in analyzing time-series data and capturing temporal dependencies in ECG signals. When combined with CNNs, hybrid models can achieve accuracies exceeding 98.7% [3][4].

D. Convolutional Neural Networks (CNN)

CNNs are the most effective models for ECG classification, achieving >99% accuracy in arrhythmia detection by automating feature extraction [2]. Their scalability and ability to handle raw signals make them the gold standard for clinical applications [8][10].

The review of machine learning methodologies for classifying ECG signals demonstrates significant advancements in automated cardiac abnormality detection. Among the explored techniques, CNNs emerged as the most effective due to their superior ability to handle raw signal data, automate feature extraction, and achieve high classification accuracy, often exceeding 99%. While models such as SVMs and ANNs provide valuable insights, their reliance on feature engineering or limited scalability poses challenges for large-scale ECG datasets. Hybrid models, like CNN-LSTMs, show promise by combining spatial and temporal analysis capabilities, achieving notable accuracy improvements. Despite these advancements, challenges such as dataset imbalance, variability in ECG signals, and computational complexity remain significant hurdles. Addressing these limitations and exploring novel techniques, such as transfer learning and real-time implementations, could pave the way for further integration of machine learning in cardiovascular diagnostics. This review highlights the transformative potential of machine learning in improving healthcare outcomes through efficient and accurate ECG analysis.

III. Methodology

This section talks about the steps used to process the ECG signal, extract essential parameters to generate synthetic ECG signals, and training a machine learning model to classify heart abnormalities using the synthetic ECG signals.

A. Data Collection and Preprocessing

The ECG signals used in this study were from the MIT-BIH Normal Sinus Rhythm Database [13]. The signal was sampled at a 128 Hz frequency, providing sufficient resolution to capture essential cardiac features. Convolution Neural Network (CNN) was chosen as the machine learning model to classify the ECG signals, due to its remarkable accuracy in classifying ECG signals.

The raw signals were processed to prepare for parameter extraction to generate a comprehensive dataset of synthetic ECG signals for both normal and abnormal cardiac conditions, including Premature Ventricular Contraction (PVC), Atrial Fibrillation (AFib), and Ventricular Tachycardia (VTach), to train the CNN machine learning model.

Signal Filtering: The signal was processed using a three-stage filtering process to remove the baseline wander, high-frequency noise, and powerline interference. A second-order Butterworth high-pass filter with a cutoff frequency of 0.5Hz [14] removes baseline wander, which is a low-frequency artifact often caused by breathing and electrode movement during ECG measurement. Afterward, a second-order Butterworth low-pass filter was used with a cutoff frequency of 40Hz [14] in order to remove high-frequency noise that caused electrical interference and muscle activity. Then, an IIR notch filter was applied with a central frequency of 50Hz [14] and a bandwidth of 1Hz [14] to eliminate powerline interference.

Parameters Extraction: Various parameters were extracted from the filtered ECG signal including P-wave, T-wave and QRS Complex amplitudes, P-wave and T-wave frequencies and phase shifts, QRS complex width and time of occurrence. These parameters are crucial to model the synthetic signals.

1. QRS Complex Amplitude: The QRS complex amplitude was found by finding the average of the total R-peaks found using MATLAB's *findpeaks* function for each signal; the function finds the R-peak, which is the most prominent feature in the QRS Complex. The Minimum peak height was set to 0.5mV to exclude small noise peaks [15], and the Maximum Peak height was set to 0.6 ´ the sampling frequency, which is equivalent to 600ms, which corresponds to a maximum normal heart rate of 100 bpm to prevent false positives [16] and the location in samples for each peak was found simultaneously using the same function.
2. P-wave and T-wave amplitude: The amplitude of the P and T wave was found relative to the position of the R-peak [16] by predefining a detection window of (R-peak location in samples - 0.2 ´ the sampling frequency) to (R-peak location in samples – 0.1 ´ the sampling frequency) the P-wave amplitude can be found using the *findpeaks* MATLAB function. The same goes when finding the T-wave amplitude but with a detection window of (R-peak location in samples + 0.2 ´ the sample frequency) to (R-peak location in samples + 0.4 ´ the sample frequency) [16]. Then, the Average peak was taken for both the P-wave and T-wave for each signal.
3. P-wave and T-wave frequency: The average frequency of both the P-wave and T-wave was calculated based on the interval of the successive peaks.
$$f_P = \frac{1}{Average\ P\ Interval}$$
The same equation was used to find the T-wave frequency. [14].

4. P-wave and T-wave Phase Shift: The phase shifts of the P-wave and T-wave were found relative to the R-peak, and it was calculated using: $\phi=2\pi f\Delta t$ where f is the frequency of the wave either the P-wave or T-wave and the Dt is the time difference between the R-peak and the wave's peak [17]. Then, the average was found for each wave's phase shift for each signal in the dataset.
5. QRS Complex Timing: The timing of the QRS complex was found as the time of the R-peak in the QRS complex.
$$t_q = \frac{Sample\ Indices\ of\ Detected\ R-Peak}{Sampling\ Frequency}$$
t_q here represents the time at which the QRS complex occurred. [17]

6. QRS Complex Width: A baseline defined as the average amplitude of the signal during the first 0.5s, it serves as a reference point to identify significant deviations in the signal. Then, the width of the QRS complex was determined as the time interval between the start and end of the QRS complex, where the start is the first point before the R-peak where the signal deviates by more than 30% of the R-peak amplitude relative to the baseline, and the same goes to finding the endpoint [17]. Finding the values of these parameters will be used in the mathematical equation to model the synthetic ECG signal.
7. Generating the Synthetic ECG signals: Synthetic ECG signals for normal and abnormal cardiac conditions, including Premature Ventricular Contraction (PVC), Atrial Fibrillation (AFib), and Ventricular Tachycardia (VTach), Were modeled using mathematical representation of the P-wave, T-wave, and QRS complex to generate the dataset to train the machine learning model to classify normal and abnormal ECG signals. The synthetic ECG was generated by summing the individual representation of each wave.

Normal ECG Model.

$$ECG(t) = P(t) + QRS(t) + T(t) \quad (1)$$

Where:

$$P(t) = A_P sin(2\pi f_P t + \Phi_P) \quad (2)$$

Which is the mathematical representation of the P-wave.

Where:

A_P: P-wave amplitude

f_P: P-wave frequency

Φ_P: P-wave Phase Shift

$$QRS(t) = A_Q e^{(-\frac{(t-t_Q)^2}{2\sigma_Q^2})} \qquad (3)$$

Which is the mathematical representation of the QRS Complex.
Where:

A_Q: QRS Complex amplitude

t_Q: QRS Complex timing

σ_Q: QRS Complex width

$$T(t) = A_T sin(2\pi f_T t + \Phi_T \qquad (4)$$

Which is the mathematical representation of the T-wave.

Where:

A_T: T-wave amplitude

f_T: T-wave frequency

Φ_T: T-wave Phase Shift.

Premature Ventricular Contraction (PVC):

$$ECG_{PVC}(t) = QRS_{PVC}(t) + T(t) \qquad (5)$$

Where:

$$QRS_{PVC}(t) = 1.5 \times A_Q e^{(-\frac{(t-t_Q)^2}{2\sigma_Q^2})} \qquad (6)$$

Where:

$P(t) = 0$

Atrial Fibrillation (AFib):

$$ECG_{AFib}(t) = QRS(t) + T(t) \qquad (7)$$

Where:

$P(t) = 0$

Ventricular Tachycardia (VTach):

$$ECG_{VTach}(t) = QRS_{VTach}(t) \qquad (8)$$

Where:

$$QRS_{VTach}(t) = 1.8\times A_Q e^{(-\frac{(t-t_Q)^2}{2\sigma_Q^2})} \qquad (9)$$

$P(t) = 0 \quad \& \quad T(t) = 0$

Using MATLAB, the parameters found from processing the signal were plugged into the equations, and a database of 40 synthetic signals was generated, with each condition having ten different generated signals.

Dataset Preparation: The dataset used for training, validating, and testing the model was the 40 synthetic ECG signals that were previously generated, complemented with the individual wave component for each signal. The 40 synthetic signals were split into 70% for training, 15% for validation, and 15% for testing, ensuring stratified samples to maintain class balance [18], [19]. Then, all the signals were processed to consist of 1280 samples to maintain consistency in the dataset [22].

B. Machine Learning Model

In this section, the machine learning model used to classify the ECG signal will be described and evaluated.

Model Architecture: A multi-input Convolution Neural Network (CNN) was designed to process each full ECG signal and its individual components simultaneously. This approach allows the model to process each component individually and extract specialized features from each. Feeding the model the separated wave components prevents the model from diluting the wave's individual characteristics during feature extraction. This enhances the ability of the network to identify these patterns when classifying ECG signals [20], [23]. The architecture includes:

1. Input layers: Separate input layers were used for each of the wave components and the full synthetic ECG signal with a shape of [1280,1] [23].
2. Convolution Layers: Convolution layers are ideal for time-series data; they are efficient in extracting spatial (temporal) features, like peaks, waveform shapes, and slopes. Therefore, each of the inputs passes through a convolution layer to ensure the features of each individual input are independently

learned by the model. Each of the wave components passes through a (Conv1D) with 16 filters, a size five kernel, and ReLU activation [20],[24], while the full ECG signal passes through a (Conv1D) with 32 filters for better feature extraction [24].

3. Feature combination: Features from each input were combined into a unified feature representation, taking this hybrid approach to allow the model to identify large scale abnormalities and the subtle abnormalities in the wave components [20], [26].
4. Temporal Analysis: Long-short-term memory (LSTM) is suitable for long-term dependencies in sequential data; therefore, using them for ECG signals that are time-dependent and temporal relationships are important for diagnosis is suitable. Adding an LSTM layer enhances the ability of the model to detect patterns over time, which would be helpful in detecting irregular rhythms in the signal. An LSTM layer with 63 units was used to capture the temporal dependencies in the combined feature [23].
5. Dense Layer: Fully connected layers with 128 neurons, a dropout regularization of 50% and an L2 regularization were applied to reduce overfitting by promoting generalization and discouraging reliance on specific neurons [25], [26]. Enhancing the models performance on unseen data.
6. Output Layer: A softmax output layer was used to classify signals into one of the predefined categories by giving a probability for each class, which is suitable for multi-class classifications. It ensures precision and interpretability in class detection, which is critical for medical applications.[19].

Training Configurations: The model was combined using a Stochastic Gradient Descent SGD optimizer with momentum, with a learning rate of 0.001, and categorical cross-entropy as the loss function. These were the training configurations:

1. Callbacks: Early stops and learning rate reduction on plateau with a factor of 0.2 were used to stabilize the training [26], [27].

2. Hyperparameters: The Number of epochs was 30 with a batch size of 16.

Model Training and Validation: The model was trained on a training set and validated using the previously generated validation set.

IV.Results and Interpretation

A. Testing and Validation:

The testing was done on the previously generated testing set; the test yielded a final accuracy of 33.33% and a validation loss of 1.5708.

B. Testing and Evaluation

The model was tested on unseen data, giving us these results:

Accuracy	33.33%
Precision	0.11
Recall	0.33
F1-Score	0.17

Table 1

The detailed class-wise performance metrics are shown in Table 2.

Class	*Precision*	*Recall*	*F1-Score*	*Support*
0	*0.00*	*0.00*	*0.00*	*1*
1	*0.33*	*1.00*	*0.50*	*2*
2	*0.00*	*0.00*	*0.00*	*1*
3	*0.00*	*0.00*	*0.00*	*2*

Table 2

Table 2 shows that the model mainly predicted Class 1 while failing to classify other classes accurately. This suggests significant class imbalance and limited feature extraction for specific ECG signals.

The results show that the models need further optimization, a larger dataset to achieve higher results, a smaller number of parameters and less complex ECG signals to improve performance.

VIII. Conclusion

This review highlights the exceptional performance of Convolutional Neural Networks (CNNs) in ECG classification. By automating feature extraction and achieving high accuracy, CNNs have set a new standard for ECG analysis. Addressing challenges like dataset imbalance and signal variability will further refine these systems. Future research should focus on hybrid models, transfer learning, and real-time deployment to enhance clinical integration. CNN-based systems have the potential to revolutionize cardiovascular diagnostics, improving healthcare outcomes worldwide [2][3][8].

References

[1] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, “Cardiologist-level arrhythmia detection with convolutional neural networks,” *Nature Medicine*, vol. 25, no. 1, pp. 65–69, 2019.

[2] M. A. Khan, M. S. Hossain, and M. Riaz, “Automated detection of myocardial infarction using machine learning techniques on ECG signals,” *Computers in Biology and Medicine*, vol. 101, pp. 341–354, 2018.

[3] Y. Zhang, Y. Liu, and X. Wang, “A hybrid approach for the classification of ECG signals using machine learning and signal processing techniques,” *Computers in Biology and Medicine*, vol. 122, p. 103800, 2020.

[4] R. Kumar, A. Gupta, and P. Sharma, “Hybrid CNN-LSTM for ECG signal classification with wavelet feature extraction,” *Neural Computing and Applications*, vol. 33, no. 16, pp. 8121–8131, 2021.

[5] A. Ahamed, M. M. Rahman, and M. K. Hasan, “Long short-term memory networks for ECG classification,” *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 12, pp. 3450–3460, 2020.

[6] M. Alim and M. Islam, “Application of support vector machines for ECG signal classification,” *Biomedical Signal Processing and Control*, vol. 55, p. 101619, 2020.

[7] N. Strodthoff, P. Wagner, T. Schaefer, and W. Samek, "Deep learning for ECG analysis: Benchmarks and insights from PTB-XL," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 8, pp. 2362–2372, 2020.

[8] Z. Zhao, S. Qiu, and Y. Li, “ECG classification using convolutional neural networks with global average pooling,” *Biomedical Signal Processing and Control*, vol. 63, p. 102144, 2021.

[9] X. Zhou, Z. Li, and H. Peng, “Transfer learning for ECG classification using pre-trained deep neural networks,” *Frontiers in Physiology*, vol. 11, p. 954, 2020.

[10] M. Hassaballah, M. Fawzy, and M. Khalil, “Optimization techniques for ECG classification using metaheuristic algorithms,” *Expert Systems with Applications*, vol. 175, p. 114837, 2021.

[11] A. Goldberger et al., “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[12] S. Sudila and R. Poravi, “Deep residual CNNs for ECG signal classification,” *Biomedical Signal Processing and Control*, vol. 66, p. 102478, 2021.

[13] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark, ... and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. [Online]. Available: https://doi.org/10.1161/01.CIR.101.23.e215

[14] GE Healthcare, "A guide to ECG signal filtering," Insights, [Online]. Available: https://www.gehealthcare.com/insights/article/a-guide-to-ecg-signal-filtering. [Accessed: Nov. 22, 2024]

[15] Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," Circulation, vol. 101, no. 23, pp. e215–e220, 2000. [Online]. Available: https://doi.org/10.1161/01.CIR.101.23.e215

[16] GE Healthcare, "A guide to ECG signal filtering," Insights, [Online]. Available: https://www.gehealthcare.com/insights/article/a-guide-to-ecg-signal-filtering. [Accessed: Nov. 22, 2024].
[17] Pan, J., and Tompkins, W. J., "A Real-Time QRS Detection Algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.
[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016.
[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 25, pp. 1097–1105, 2012.
[21] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.
[22] A. Goldberger, et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
[24] F. Chollet, *Deep Learning with Python*, Shelter Island, NY, USA: Manning Publications, 2017.
[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[26] N. Srivastava, et al., "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.
[27] G. E. Hinton, et al., "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.