



Deep-Learning-Driven Heartbeat Classification Using Segmented and Preprocessed ECG Signals

Ali Husham Hameed, Ali Mokdad

Computer Science, Faculty of Science & Literature, American University of Culture and Education, Beirut, Lebanon

Article Info

Article history:

Received 03, 12, 2025
Revised 05, 10, 2025
Accepted 07, 02, 2025
Published 10, 30, 2026

Keywords:

Deep Learning,
ECG Classification,
Neural Networks,
Attention Mechanism,
Heartbeat Detection,
Cardiovascular Diseases.

ABSTRACT

Cardiovascular diseases are a leading cause of death worldwide, so accurate and timely diagnosis is essential for effective treatment. Electrocardiogram (ECG) signals offer valuable information for detecting various cardiac abnormalities. Traditional ECG analysis is time-consuming and prone to human error, particularly in distinguishing between different types of arrhythmias. Manual interpretation of large amounts of ECG data can lead to misdiagnosis or delayed treatment. There is an urgent need for a reliable solution capable of automatically classifying recorded heart rhythms with high accuracy. This study proposes a lightweight yet effective 1D convolutional neural network (CNN) enhanced with both channel and temporal attention mechanisms. These attention modules enable the model to focus selectively on the most informative features across the heartbeat waveform. The model was trained and evaluated on the MIT-BIH arrhythmia dataset, comprising over 100,000 heartbeat samples categorized into five classes. A class-weighted cross-entropy loss function and learning rate scheduling were employed to handle data imbalance and improve convergence. The proposed model achieved a validation accuracy of 99% and a test accuracy of 98%. This study presents a reliable and computationally efficient solution for real-time ECG classification tasks.

This is an open access article under the [CC BY](#) license.



Corresponding Author:

Ali Husham Hameed

Computer Science, Faculty of Science & Literature,
American University of Culture and Education, Beirut, Lebanon

Email: Alihusham83@yahoo.com



1. INTRODUCTION

The human body quietly leaks a stream of tiny electrical signals that doctors now use to keep tabs on health. An electrocardiogram, or ECG, listens to those heart sparks and is probably the best-known test for checking a beating rhythm [1]. Its tracing quickly reveals whether the heart pushes blood the way it should or stumbles along crooked. In the same spirit, electromyography, or EMG, picks up the chatter of muscles when they flex, whether in an arm, a leg, or even the jaw [2]. Because doctors can read those traces almost in real time, they often step in before a small trouble balloon into a bigger crisis. Smart, self-learning programs now build on that idea, sifting through the signals automatically and flagging readings that matter most.

An ECG reads the hearts electrical rhythm and alerts doctors to several linked problems at once, so using machine-learning methods to sort those traces adds real punch to modern health apps [3]. When a computer flags an issue early, care teams can move quicker, an edge that really matters as heart-attack numbers keep climbing [4]. Yet even seasoned cardiologists can struggle with rhythm quirks-browsing hours of strip data takes ages and tiny errors can slip through [5]. Gadgets like Holter monitors help, yet readings captured far from the clinic can be fuzzy, noisy, and streaky with interference. Throw in petabytes of files from hospitals and wearables alike, and crafting a sturdy, turn-the-key arrhythmia-detector becomes a heavy lift for anyone in bio-signal work.

Lately, machine learning algorithms and deep neural networks have stormed that arena, chewing through heart traces and handing clinicians spot-on labels or red flags in record time [6]. Traditional ML algorithms such as Support Vector Machines (SVM) and Random Forests (RF) have been widely used for feature-based ECG

classification, while DL architectures like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have demonstrated superior performance by learning hierarchical features directly from raw signals [7]. These advances are central to real-time health monitoring and have seen growing integration in wearable and clinical technologies.

Several studies have applied DL to ECG signal classification with promising results. For instance, Kachuee et al. (2018) [8] achieved 95.9% and 93.4% accuracy in myocardial infarction and arrhythmia detection, respectively, using a deep CNN on Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) and Physikalisch-Technische Bundesanstalt (PTB) datasets. Similarly, Mathews et al. (2018) [9] used Deep Belief Networks (DBN) and Restricted Boltzmann Machines (RBM) to classify ventricular and supraventricular beats with high accuracy despite a low sampling rate. Alfaras et al. (2019) [10] introduced an efficient model using Echo State Networks (ESN) that required only a single ECG lead, showing promise for wearable device integration. This model achieved a sensitivity of 92.7% and a predictive value of 86.1% on ARMITB and American Heart Association (AHA) databases.

Liang et al. (2020) [11] compared a Bidirectional Long Short-Term Memory (BiLSTM)-based CNN model with an evolutionary neural network approach. While Method I required longer training time, it provided superior classification stability. Weimann and Conrad (2021) [12] emphasized transfer learning to reduce annotation needs by 80%, demonstrating the utility of pretraining in ECG tasks. More recently, Tang et al. (2024) [13] applied metaheuristic optimization to ML classifiers, improving performance through parameter tuning. Desmalia et al. (2024) [14] combined principal component analysis (PCA) with ML models like K-Nearest Neighbor (k-NN), RF, and SVM, reporting 86% accuracy using k-NN. Yahya and Edan (2025) [15] employed a hybrid LSTM-CNN model and outperformed other approaches with 97.25% accuracy across 29 heart conditions.

Attention mechanisms enhance time series analysis by allowing models to focus on the most informative parts of the input sequence. Channel attention, such as Squeeze-and-Excitation (SE) blocks [16], recalibrates feature maps by emphasizing important channels across time. Temporal (spatial) attention focuses on specific time steps in the sequence, useful for identifying key signal intervals [17]. Hybrid attention combines both, enabling models to learn what features matter and when they matter in the sequence—crucial for ECG signal classification [18]. Self-attention, as in Transformers, models long-range dependencies by letting each time step attend to all others, providing a global view of the sequence [19]. Lightweight self-attention variants can be embedded in CNNs for efficiency. Together, these mechanisms improve model interpretability, robustness, and performance in complex biomedical tasks.

Previous studies on ECG classification have primarily focused on standard deep learning models often lacking advanced mechanisms to capture both spatial and temporal dependencies. While some research introduced channel or temporal attention separately, few have explored their combined application within a unified architecture. This research aims to design a deep learning model enhanced with both channel and temporal attention mechanisms for accurate ECG arrhythmia classification. The goal is to improve feature learning and robustness on imbalanced and noisy datasets.

Material and Methods

This research investigates the automated classification of Heartbeat utilizing ECG waveform data. The ECG signal classification methodology using deep learning techniques is presented in Figure 1.

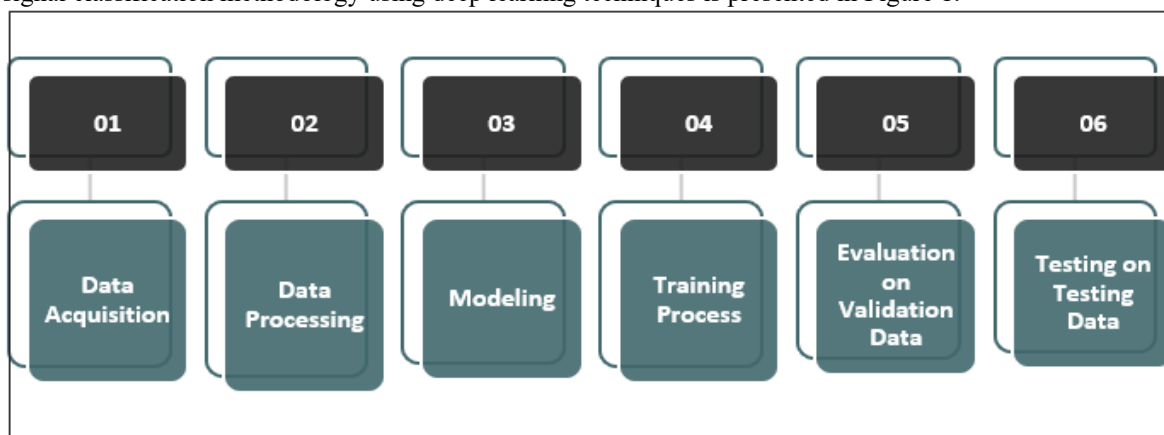


Figure 1: Main steps of the proposed methodology.

It begins with data collection and processing, followed by designing and training a robust deep learning-based model. Performance evaluation on both validation and independent test sets ensures reliability and generalizability. Google Colab is a platform for executing Python scripts because of its GPU support, which is

beneficial for deep learning tasks. For data processing and analysis, the pandas and numpy libraries are imported. Deep learning frameworks such as PyTorch are used to build, train, and evaluate neural network models. Scikit-learn (sklearn) is also used to partition the dataset and evaluate the model.

Data Acquisition

The dataset from Kaggle [20], titled ECG Heartbeat Categorization Dataset, was chosen because of its well-organized balance, which is perfect for deep learning endeavors. It merges two well-established datasets: the MIT-BIH Arrhythmia Dataset and the PTB Diagnostic ECG Database, both obtained from PhysioNet. The dataset contains segmented ECG signals that have been preprocessed, each corresponding to one heartbeat. Preprocessing was done by down-sampling to 125Hz and zero-padding so the samples would have a uniform length of 188. Each heartbeat is marked, which makes it easier to utilize supervised learning to differentiate between normal and abnormal cardiac activities.

From the MIT-BIH Arrhythmia Database there is a total of 109,446 samples available, which are distributed over five categories. The PTB Diagnostic ECG Database supplies 14,552 samples with two categories, mostly intended for use in distinguishing healthy from sick heart conditions. The dataset is saved in several CSV files containing all data arranged in tabular format. Each row represents a single heartbeat segment, and the last entry in each row is the class label.

The first few rows shown in Figure 2 display real-valued inputs, with many zeros at the end due to padding, ensuring that all segments are of equal length for deep learning model compatibility.

	0	1	2	3	4	5	6	7	8	9	...	177	178	179	180	181	182	183	184	185	186
0	0.977941	0.926471	0.681373	0.245098	0.154412	0.191176	0.151961	0.085784	0.058824	0.049020	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.960114	0.863248	0.461538	0.196581	0.094017	0.125356	0.099715	0.088319	0.074074	0.082621	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	1.000000	0.659459	0.186486	0.070270	0.070270	0.059459	0.056757	0.043243	0.054054	0.045946	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.925414	0.665746	0.541436	0.276243	0.196133	0.077348	0.071823	0.060773	0.066298	0.058011	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.967136	1.000000	0.830986	0.586854	0.356808	0.248826	0.145540	0.089202	0.117371	0.150235	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5 rows × 187 columns																					

Figure 2: The first rows of the dataset.

The training data was loaded into the data frame from the CSV file mitbih_train.csv, without specifying a header row. The target labels were extracted from the last column of the data frame and converted to an integer type for compatibility with machine learning models. Training data features were extracted from all but the last columns in a data frame, resulting in 187 numerical features for each heartbeat sample. These values represent the amplitudes of the pre-processed and calibrated ECG signal corresponding to each heartbeat segment.

Despite potential concerns about data leakage, our study carefully handles the dataset by strictly separating training and testing sets to minimize overlap and ensure fair evaluation. The use of pre-segmented, uniformly processed ECG signals allows consistent input for deep learning models, enhancing reproducibility.

Data Processing

The dataset undergoes several preprocessing steps to prepare it for model training:

- **Conversion of Features to Float Type:** Data processing begins by converting all feature values to a float data type, which ensures numerical accuracy and consistency because deep learning models require inputs in consistent and continuous numerical formats. The dataset contains five unique heartbeat classes represented by labels, each corresponding to a specific type of ECG signal. These classes are denoted as: 'N' (normal), 'S' (supraventricular ectopic beat), 'V' (ventricular ectopic beat), 'F' (fusion beat), and 'Q' (unknown beat), mapped to numeric labels 0 through 4. Table 1 highlights the imbalance in class distribution within the ECG dataset, where the majority of samples (72,471) belong to the 'Normal' class. In contrast, minority classes like 'Fusion' and 'Supraventricular' have significantly fewer samples, posing a challenge for model training and requiring techniques to address class imbalance.

Table 1: Distribution of samples into the five classes.

Class Label	Class Name	Count
0	N	72471 sample
1	S	2223 sample
2	V	5788 sample
3	F	641 sample
4	Q	6431 sample

- Reshaping: The feature matrix is transformed into a NumPy matrix containing 87,554 samples, each with 187 features. To match the expected input format for one-dimensional convolutional neural networks, it is reshaped to (87554, 1, 187), with the addition of a channel dimension. The label matrix is reshaped to (87554, 1) to ensure consistency with the training procedure.
- Normalization: The dataset (feature matrix) was normalized by calculating the mean (0.1743), standard deviation (0.2263), maximum (1.0), and minimum values of the raw signal. The means were subtracted from each data point and then divided by the standard deviation to ensure that the mean and variance were equal to unit, which is essential for effective training of neural networks. After normalization, the new mean was approximately zero, and the standard deviation was exactly one, confirming the success of the process. The label matrix is converted to an integer type to ensure compatibility with model training procedures that expect integer class labels.
- Splitting: The dataset was split into two sets: training (80%) and validation (20%). The resulting training set has a marked class imbalance, with the N (normal) class dominating with 57,965 samples, while minority classes such as F and S have much fewer samples—517 and 1,767, respectively. The validation set also reflects this distribution trend.
- Class Distribution Balancing: To handle class imbalance, class weights were computed inversely proportional to class frequencies and normalized between 0 and 1. The dataset was converted into PyTorch tensors using TensorDataset for both training and validation phases. A WeightedRandomSampler was then applied to the training DataLoader to ensure that underrepresented classes received higher selection probabilities. This approach enabled each mini-batch to include a fair distribution of all classes, improving the model's learning capability across rare and dominant heartbeat types. The technique helps reduce the risk of model bias toward the majority class, particularly the normal (N) category. However, the sampler's effectiveness relies on fine-tuning of sampling weights to prevent overfitting and maintain generalization. This results in a more balanced batch during training, as shown in Table 2. This balanced representation improves the model's ability to learn from all classes effectively.

Table 2: Class distribution with the batch size of 1024.

Class Label	Class Name	Count
0	N	208 per batch
1	S	206 per batch
2	V	209 per batch
3	F	213 per batch
4	Q	188 per batch

Modeling

The proposed model is designed to effectively capture both local and global patterns in ECG signals by integrating 1D Convolutional Neural Networks (CNNs) with attention mechanisms. The architecture utilizes Network-in-Network (NIN) blocks to enhance feature abstraction through successive 1×1 convolutions, improving model expressiveness with fewer parameters. Channel-temporal attention (CTA) modules are strategically placed after NIN blocks to allow the model to focus on the most informative features and critical time segments. Table 3 details the structure of the proposed model.

Table 3: Description of hybrid attention-based CNN model.

Layer Type	Description	Output Shape
Input	1D time-series ECG signal (187 samples)	(batch_size, 1, 187)
1st NIN Block	Conv1D (kernel=11, stride=4) → Batch Normalization (BN) → Gaussian Error Linear Unit (GELU) activation → Conv1D (1×1) → GELU → Conv1D (1×1) → GELU	(batch_size, 48, 44)
Max Pooling	1D MaxPooling (kernel=3, stride=2)	(batch_size, 48, 21)
Channel-Temporal Attention	SE-style channel attention + temporal convolutional attention	(batch_size, 48, 21)

2nd NIN Block	Conv1D (kernel=5, padding=2) → BN → GELU → Conv1D (1×1) → GELU → Conv1D (1×1) → GELU	(batch_size, 128, 21)
Max Pooling	1D MaxPooling (kernel=3, stride=2)	(batch_size, 128, 10)
Channel-Temporal Attention	Repeats CTA with 128 channels	(batch_size, 128, 10)
3rd NIN Block	Conv1D (kernel=3, padding=1) → BN → GELU → Conv1D (1×1) → GELU → Conv1D (1×1) → GELU	(batch_size, 256, 10)
Max Pooling	1D MaxPooling (kernel=3, stride=2)	(batch_size, 256, 4)
Dropout	Dropout (p = 0.4)	(batch_size, 256, 4)
Final NIN Block	Conv1D (kernel=3, padding=1) → BN → GELU → Conv1D (1×1) → GELU → Conv1D (1×1) → GELU	(batch_size, 5, 4)
Adaptive Pooling	Average Global pooling over time dimension	(batch_size, 5, 1)
Flatten	Converts output to 2D for classification	(batch_size, 5)

The model was chosen for its balance between complexity, interpretability, and performance in time series tasks like ECG analysis. Two sequential sub-modules were inserted after key convolutional blocks in the CNN model:

- Channel Attention Sub-module: It compresses temporal information via global average pooling and uses a multi-layer perceptron (MLP) with ReLU and sigmoid activations to generate channel-wise attention weights. These weights re-weight the original feature map, emphasizing important channels.
- Temporal Attention Sub-module: It generates temporal attention weights by applying global average and max pooling across channels, concatenating the results, and passing them through a 1D convolution with sigmoid activation. These weights highlight important time steps by re-weighting the channel-attended feature map.

The use of both global average and max pooling further enriches temporal feature extraction. Attention modules are placed after the second and third nin_block layers to optimally refine representations post-critical feature extraction stages.

Training Process

Table 4 provides the hyperparameter values of the model used during training. The training process began by initializing the model on a GPU with a cross-entropy loss function, incorporating class weights to address imbalance. An Adam optimizer and StepLR scheduler were employed to manage learning dynamics over 100 epochs.

Table 4: Hyperparameter settings for training the hybrid attention-based CNN mode.

Hyperparameter	Value
Learning Rate	0.001
Learning Rate Decay	0.8
Batch Size	64
Epochs	100
Optimizer	Adam
Loss Function	Weighted Cross Entropy
Weight Decay	0.0001

Early training phases saw significant improvements, with validation accuracy jumping from 6.6% in epoch 1 to over 85% by epoch 2, and steadily rising thereafter. The model consistently improved, reaching over 98% validation accuracy by epoch 19. The loss curves, in Figure 3, visually confirm the steady decrease in both training and validation loss. This indicates strong convergence and effective generalization. The final model achieved high performance with only 1.17MB size and 307,643 trainable parameters, reflecting a compact yet powerful architecture.

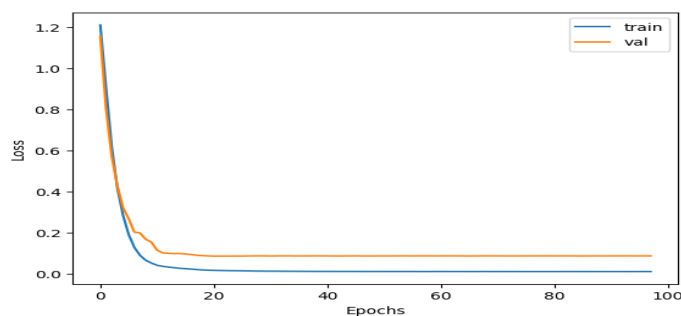


Figure 3: Training and validation loss over epochs.

The best-performing model parameters are loaded and saved to a file. The evaluation results were analyzed using four commonly used evaluation techniques, which were combined to evaluate the overall performance of the model:

- Accuracy measures the overall correctness of the model by calculating the proportion of true predictions out of total predictions.
- Precision evaluates how many of the predicted positive cases are actually correct, reflecting the model's ability to avoid false positives.
- Recall, also known as sensitivity, determines how many actual positive cases the model correctly identifies, indicating its effectiveness in detecting true positives.
- F1-score provides a balance between precision and recall, especially useful when dealing with imbalanced data, by computing their harmonic mean.
- Confusion matrix provides a comprehensive view of the model's performance.

Evaluation on Validation Data

The model was evaluated, and the results were visualized using a confusion matrix, as shown in Figure 4. The confusion matrix plotted for the validation set reveals excellent model performance, especially on the majority class 'N' (Normal) and other well-represented classes like 'V' and 'Q'.

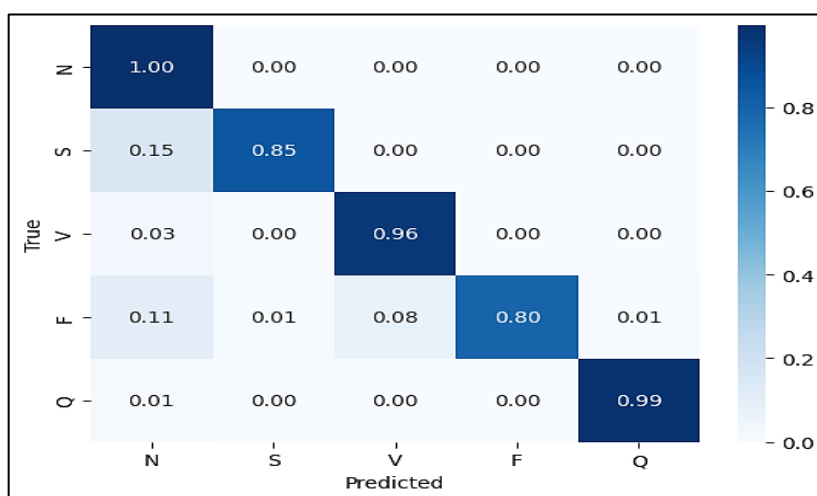


Figure 4: Confusion matrix of validation set.

The validation results in Figure 5 demonstrate the model's strong performance, achieving an overall accuracy of 99%. High precision and recall across most classes indicate effective generalization, with slight dips in minority class predictions.

	precision	recall	f1-score	support
N	0.99	1.00	0.99	14479
S	0.91	0.85	0.88	468
V	0.98	0.96	0.97	1161
F	0.88	0.80	0.83	123
Q	0.99	0.99	0.99	1280
accuracy			0.99	17511
macro avg	0.95	0.92	0.93	17511
weighted avg	0.99	0.99	0.99	17511

Figure 5: Validation results.

Testing on Testing Data

The test dataset, containing 21,892 ECG samples each with 187 features, was loaded and preprocessed to align with the model's input format. The final column was extracted as the label, while the remaining columns were treated as features and reshaped to a (samples, 1, 187) format for compatibility with 1D CNN input. Data normalization was applied using z-score scaling to ensure consistent feature distribution. Labels were reshaped for compatibility and converted to PyTorch tensors along with the input data. Finally, a Tensor Dataset and Data Loader were constructed to facilitate efficient batch-wise evaluation. The previously trained and saved model was loaded and evaluated on the test dataset. To interpret performance, a confusion matrix was plotted, shown in Figure 6, which provides a visual summary of the classification results across all five categories.

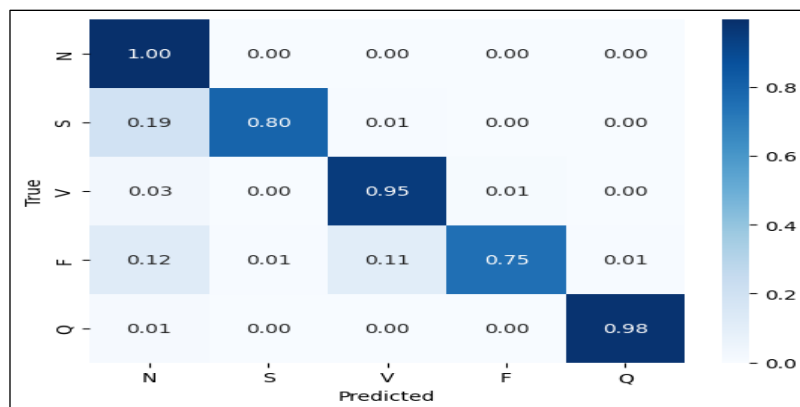


Figure 6: Confusion matrix of the testing set.

As shown in Figure 7, the model demonstrated robust generalization on the testing dataset. It achieved an impressive 98% overall accuracy across 21,892 ECG segments. Normal beats (N) were classified with near-perfect precision and recall, while ventricular (V) and unknown (Q) beats also yielded high F1-scores of 0.96 and 0.99, respectively. The supraventricular (S) and fusion (F) beats—typically harder to distinguish due to their subtle patterns—achieved F1-scores of 0.84 and 0.80. The macro average F1-score of 0.91 and macro recall of 0.90 indicate that the model effectively balanced performance across all heartbeat categories. This confirms the model's capability to detect rare arrhythmias with high reliability, making it suitable for real-world ECG classification tasks.

	precision	recall	f1-score	support
N	0.99	1.00	0.99	18118
S	0.88	0.80	0.84	556
V	0.97	0.95	0.96	1448
F	0.85	0.75	0.80	162
Q	0.99	0.98	0.99	1608
accuracy			0.98	21892
macro avg	0.94	0.90	0.91	21892
weighted avg	0.98	0.98	0.98	21892

Figure 7: Testing results.

Results Discussion

The proposed model introduces a novel design by integrating channel and temporal attention sequentially within a unified module for 1D CNNs, enhancing the capture of both feature importance and temporal relevance—crucial for ECG classification. Unlike prior methods that treat these attentions separately, our approach offers a more comprehensive understanding of ECG signals. As shown in Table 5, the model achieves strong performance, outperforming previous studies using the same data. These results validate its effectiveness and robustness in accurately classifying ECG data.

Table 5: Comparison with previous studies.

Reference	Model	Results
Tang et al. (2024) [13]	XGBoost and metaheuristic algorithm (JADE)	Accuracy: 86% Precision: 89% Recall: 86% F1-Score: 87%
Desmalia et al. (2024) [14]	K-Nearest Neighbor (k-NN)	Accuracy: 86% Precision: 86% Recall: 91% F1-score: 88%
Yahya and Edan (2025) [15]	Hybrid deep learning model (LSTM-CNN)	Accuracy: 97.25%
Our Method	Hybrid Attention-based CNN model	Accuracy: 98% Precision: 94% Recall: 90% F1-Score: 91%

The results demonstrate that the proposed attention-enhanced deep learning model achieved outstanding performance, with 99% validation accuracy and 98% test accuracy, indicating its strong potential for automated ECG heartbeat classification. These outcomes highlight the model's ability to effectively learn both dominant and subtle patterns in imbalanced clinical data, especially through the integration of temporal and channel attention mechanisms.

The exceptionally high validation and test accuracies can be partly attributed to the model's effective use of attention mechanisms and class-weighted loss, which likely enhanced its focus on informative features and addressed class imbalance. Preprocessing and careful architectural choices may have further improved learning stability and feature discrimination. However, since both validation and test sets are derived from the same data source, the model may have captured dataset-specific patterns rather than generalizable traits. This raises concerns of overfitting, emphasizing the need for external validation.

Conclusion

This study introduced a novel deep learning architecture for ECG heartbeat classification, integrating both channel and temporal attention mechanisms within a compact and efficient 1D CNN framework. The model demonstrated outstanding performance on the MIT-BIH dataset, achieving 99% validation accuracy and 98% test accuracy. The attention modules significantly enhanced the model's ability to focus on informative temporal regions and critical morphological features.

The approach also ensured efficient training and convergence, while the high performance is encouraging, further validation on unseen and more diverse datasets is essential. Overall, the proposed model holds significant potential for real-world deployment in intelligent ECG monitoring systems.

Source of Funding: No finding source.

Conflict of Interest: No conflicts of interest have been disclosed by the authors.

References

- [1] K. Husain, M. S. Mohd Zahid, S. U Hassan, S. Hasbullah, S. Mandala, Advances of ECG sensors from hardware, software and format interoperability perspectives, *Electronics*, 10, 105 (2021), DOI: [10.3390/electronics10020105](https://doi.org/10.3390/electronics10020105)
- [2] F. Sadikoglu, C. Kavalcioglu, B. Dagman, Electromyogram (EMG) signal detection, classification of EMG signals and diagnosis of neuropathy muscle disease, *Procedia Computer Science*, 120, 422-429 (2017), DOI: [10.1016/j.procs.2017.11.267](https://doi.org/10.1016/j.procs.2017.11.267)
- [3] S. Irfan, N. Anjum, T. Althobaiti, A. A. Alotaibi, A. B. Siddiqui, N. Ramzan, Heartbeat classification and arrhythmia detection using a multi-model deep-learning technique, *Sensors*, 22, 5606 (2022), DOI: [10.3390/s22155606](https://doi.org/10.3390/s22155606)
- [4] E. E. Tripoliti, T. G. Papadopoulos, G. S. Karanasiou, K. K. Naka, D. I. Fotiadis, Heart failure: diagnosis, severity estimation and prediction of adverse events through machine learning techniques, *Computational and Structural Biotechnology Journal*, 15, 26-47 (2017), DOI: [10.1016/j.csbj.2016.11.001](https://doi.org/10.1016/j.csbj.2016.11.001)
- [5] B. S. Chandra, C. S. Sastry, S. Jana, Robust heartbeat detection from multimodal data via CNN-based generalizable information fusion, *IEEE Transactions on Biomedical Engineering*, 66, 710-717 (2019), DOI: [10.1109/TBME.2018.2854696](https://doi.org/10.1109/TBME.2018.2854696)
- [6] B. Feng, T. Yu, H. Wang, K. Liu, W. Wu, W. Long, Machine learning and deep learning in biomedical signal analysis, *Frontiers in Human Neuroscience*, 17, 1183840 (2023), DOI: [10.3389/fnhum.2023.1183840](https://doi.org/10.3389/fnhum.2023.1183840)
- [7] A. Khan, A. Sohail, U. Zahoor, A. S. Qureshi, A survey of the recent architectures of deep convolutional neural networks, *Artificial Intelligence Review*, 53, 5455-5516 (2020), DOI: [10.1007/s10462-020-09825-6](https://doi.org/10.1007/s10462-020-09825-6)

- [8] M. Kachuee, S. Fazeli, M. Sarrafzadeh, ECG heartbeat classification: A deep transferable representation, In *2018 IEEE International Conference on Healthcare Informatics (ICHI) (IEEE, 2018)*, pp. 443-444, DOI: [10.1109/ICHI.2018.00074](https://doi.org/10.1109/ICHI.2018.00074)
- [9] S. M. Mathews, C. Kambhamettu, K. E. Barner, A novel application of deep learning for single-lead ECG classification, *Computers in Biology and Medicine*, 99, 53-62 (2018), DOI: [10.1016/j.compbiomed.2018.06.008](https://doi.org/10.1016/j.compbiomed.2018.06.008)
- [10] M. Alfaras, M. C. Soriano, S. Ortín, A fast machine learning model for ECG-based heartbeat classification and arrhythmia detection, *Frontiers in Physics*, 7, 103 (2019), DOI: [10.3389/fphy.2019.00103](https://doi.org/10.3389/fphy.2019.00103)
- [11] Y. Liang, S. Yin, Q. Tang, Z. Zheng, M. Elgendi, Z. Chen, Deep learning algorithm classifies heartbeat events based on electrocardiogram signals, *Frontiers in Physiology*, 11, 569050 (2020), DOI: [10.3389/fphys.2020.569050](https://doi.org/10.3389/fphys.2020.569050)
- [12] K. Weimann, T. O. Conrad, Transfer learning for ECG classification, *Scientific Reports*, 11, 5251 (2021), DOI: <https://doi.org/10.1038/s41598-021-84374-8>
- [13] K. Tang, S. Ma, X. Sun, D. Guo, Optimizing machine learning for enhanced automated ECG analysis in cardiovascular healthcare, *Egyptian Informatics Journal*, 28, 100578 (2024), DOI: <https://doi.org/10.1016/j.eij.2024.100578>
- [14] S. Desmalia, A. M. Siregar, K. A. Baihaqi, T. Rohana, Comparison Model Optimal Machine Learning Model With Feature Extraction for Heart Attack Disease Classification, *Scientific Journal of Informatics*, 11, 485-492 (2024), DOI: <https://doi.org/10.15294/sji.v13i3.48890>
- [15] A. S. Yahya, N. M. Edan, ECG Signal Classification Using Hybrid and Non-Hybrid Learning Technologies, *International Journal on Perceptive and Cognitive Computing*, 11, 114-121 (2025), DOI: <https://doi.org/10.31436/ijpcc.v11i1.503>
- [16] J. Hu, L. Shen, G. Sun, Squeeze-and-Excitation Networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7132-7141 (2018), DOI: [10.48550/arXiv.1709.01507](https://doi.org/10.48550/arXiv.1709.01507)
- [17] D. Bahdanau, K. Cho, Y. Bengio, Neural Machine Translation by Jointly Learning to Align and Translate, arXiv preprint arXiv:1409.0473 (2014), DOI: [10.48550/arXiv.1409.0473](https://doi.org/10.48550/arXiv.1409.0473)
- [18] S. Woo, J. Park, J. Y. Lee, I. S. Kweon, CBAM: Convolutional Block Attention Module, *Proceedings of the European Conference on Computer Vision (ECCV)*, 3-19 (2018), DOI: [10.48550/arXiv.1807.0652](https://doi.org/10.48550/arXiv.1807.0652)
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, *Advances in Neural Information Processing Systems*, 30 (2017), DOI: [10.48550/arXiv.1706.03762](https://doi.org/10.48550/arXiv.1706.03762)
- [20] S. Fazeli, ECG Heartbeat Categorization Dataset [Data set], Kaggle (2018), Available online, Accessed 8 May 2025