ESP-JETA
ESP Journal of Engineering & Technology Advancements
ISSN: 2583-2646 / Volume 6 Issue 1 March 2026
Paper Id: JETA-V6I1P116 / Page No: 140-149

Research Article

A Deep Capsule Network for Five-Class ECG Rhythm Classification from 1-D Heartbeat Segments

Ali Osman SELVİ[1], Shamistan HUSEYNOV[2]
[1]*Bilecik Seyh Edebali University, Vocational School, Computer Technology, Bilecik, Turkiye.*
[1]*Bilecik Şeyh Edebali University, Department of Electronics and Computer Engineering, Bilecik, Turkiye.*

Received Date: 23 February 2026 Revised Date: 02 March 2026 Accepted Date: 29 March 2026

Abstract: *This study proposes a deep learning-based Capsule Network (CapsNet) model for five-class ECG rhythm classification using one-dimensional heartbeat segments derived from the MIT-BIH Arrhythmia Database. The analysis focused on five clinically significant rhythm classes: Normal sinus rhythm (N), supraventricular premature beat (S), premature ventricular contraction (V), fusion beat (F), and unclassifiable beat (Q). In the preprocessing stage, raw ECG recordings were segmented into fixed-length heartbeat samples of 300 points based on annotation positions. To reduce the negative impact of class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was applied before model training. The Capsule Network architecture proposed in this study is designed to allow feature extraction from ECG data while distorting the structural features of the data. Classification results were evaluated using F1 Score, Sensitivity and Recall, and Accuracy metrics. The studies showed that the proposed architecture achieved an overall success rate of 99% in both training and test datasets. Class-based sensitivity, recall, and F1-score values generally ranged from 0.98 to 1.00. Even with erroneous classification transitions between structurally similar classes, the proposed architecture demonstrated sufficient classification performance for ECG rhythm classification.*

Keywords: *Electrocardiography (ECG), Arrhythmia Classification, Deep Learning, Capsule Network.*

I. INTRODUCTION

Arrhythmias detection using electrocardiography (ECG) signals is important for differentiating cardiovascular diseases and planning timely clinical intervention. ECG-based assessment is one of the accessible and widely used diagnostic tools in this process. In clinical situations where long-term EEG recordings are required, manual evaluation by specialists can be time-consuming. Therefore, the need for computer-aided systems capable of performing automatic and reliable rhythm classification from ECG signals has become increasingly evident [1]. Considering the clinical consequences of rhythm disorders, automatic classification is not only a technical pattern recognition problem but also an important decision-support application [2,3]. There are several factors that make ECG-based arrhythmia classification difficult. The main reasons include electromagnetic interference, muscle noise, changes in electrode contact, and the fact that the signal may vary both across individuals and from beat to beat within the same individual. These factors can reduce signal quality and adversely affect classification performance [2,4]. A considerable portion of the fundamental studies on ECG arrhythmia classification has been based on signal processing methods such as windowing, filtering, and wavelet transform. In studies carried out with these methods, performance often remained sensitive to manually defined features, fixed decision rules, or selected transformation parameters. Although these methods may achieve satisfactory performance on specific datasets, they are often insufficient for handling ECG signals with different morphological characteristics. In recent years, deep learning-based architectures such as 1D CNN, U-Net, and BiLSTM have been employed to learn feature extraction directly from data, thereby partially reducing the dependency on manually designed processing steps [4–6].

Deep learning architectures achieve good results in learning the distinctive features of raw or simply pre-processed ECG signals. However, the success of deep models largely depends on the amount of available data, the evaluation strategy employed, and the extent to which the model can utilize temporal context. In small or imbalanced datasets, the risk of overfitting increases, while models focusing only on local morphological information may fail to adequately represent inter-beat rhythm relationships. Therefore, ECG classification requires consideration not only of individual waveform patterns but also of structural and temporal relationships among samples [1,2,4]. Capsule Networks have emerged as a remarkable alternative to conventional convolution-based architectures. Instead of examining a single feature, the Capsule Network architecture uses features along with their hierarchical relationships and spatial representation. The literature emphasizes that this architecture has the potential to reduce the loss of positional information caused by pooling operations and to preserve the relationships among features more consistently. In signals such as ECG, where morphological patterns contain subtle differences, this representational capability may be beneficial for multi-class rhythm discrimination [7].

The MIT-BIH Arrhythmia Dataset is one of the most commonly used benchmark resources in ECG classification research. Containing the Q, F, V, S, N classes, the dataset is widely preferred in the literature due to its clinical relevance. However, significant differences in sample counts across classes and the morphological similarity between some classes make the classification of different ECG beats more challenging [3,8–9]. It is evident from the literature that many deep learning-based approaches developed on the MIT-BIH database have achieved high performance. While one study using transfer learning-based ResNet50 and AlexNet reported an F1-score of 99.2% for a three-class structure [10], another DenseNet-based study reported an F1-score of 98.91% under imbalanced data conditions [8]. In another approach combining hybrid LSTM and 1D convolutional layers, a performance level of 98.24% was achieved across different datasets [11]. A more recent five-class LSTM study also reported results in the range of approximately 97%–98% accuracy based on patient-wise cross-validation and AAMI classification [3]. In their study, Xiang et al. demonstrated that morphological features at different levels of detail could be learned automatically using a two-level 1D CNN approach, thereby reducing the need for manual feature design [5]. In another study, the combination of U-Net and bidirectional LSTM was found to be sensitive., local morphological information and contextual sequential information were jointly utilized, achieving 98.29% classification performance in QRS detection [6]. Li et al. reported that a probabilistic approach explicitly incorporating rhythm information through RR interval distribution could provide substantial robustness, particularly in noisy environments and real-time applications [4]. According to this corpus of work, ECG analysis necessitates structures that not only attain high accuracy but also maintain structural relationships, take advantage of temporal context, and withstand noise. In this study, one-dimensional heartbeat segments from the MIT-BIH database were used to train a deep learning-based Capsule Network model for five-class ECG rhythm categorization. The study's primary goals were to assess Capsule Networks' ability to automatically extract features from 1D ECG data and to look at how well this architecture performed in differentiating five clinically significant rhythm classes.

II. MATERIALS AND METHODS

This study's suggested method was created to process ECG recordings from the MIT-BIH Arrhythmia Database in order to accomplish five-class rhythm categorization. The raw ECG signals were initially preprocessed and segregated at the heartbeat level after the database's records were integrated into the system. In this case, a standardized representation appropriate for model input was created by segmenting each heartbeat into 300 sample points. The detrimental impact of class imbalance in the dataset on classification performance was then mitigated by applying SMOTE-based balancing. A deep learning-based Capsule Network model was then fed the balanced data structure to automatically extract and classify features. In the last phase, the system generates the output over the Q, F, V, S, and N classes after classifying the input ECG segment as one of five distinct rhythm classes. Thus, the resulting pipeline presents data preparation, class balancing, deep feature learning, and multi-class decision-making steps within a single integrated framework. The block diagram illustrating the general operation of the system is presented in Figure 1.

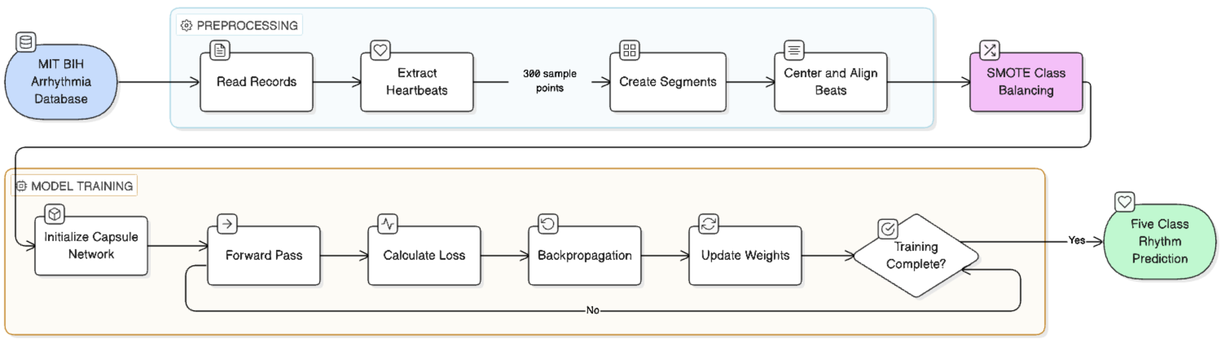

Figure 1: General block diagram of the proposed five-class ECG rhythm classification system.

A. Mit-Bih Dataset

The MIT-BIH Arrhythmia Database is frequently preferred in ECG-based arrhythmia classification applications. Due to its long-standing open-access status, the MIT-BIH Arrhythmia dataset has been used to test numerous classification algorithms. This makes it a preferred dataset for historical comparisons in current studies. In this study, the MIT-BIH Arrhythmia Database was used to test the proposed classification technique. Its widespread acceptance over the years stems from the fact that it has established a common basis for evaluation in automated arrhythmia analysis [12,13]. 48 half-hour two-channel ECG recordings from 47 people make up the MIT-BIH Arrhythmia Database. The recordings were captured with 11-bit resolution and digitized at a sample frequency of 360 Hz. During the construction of the database, the recordings were selected not only from random examples but also to include clinically important yet relatively less frequent rhythm patterns. This makes the database particularly valuable for ECG classification studies, as it allows both common classes and more difficult-to-distinguish arrhythmic examples to be examined together [12,14–15]. In this study, five main rhythm classes that cover clinically meaningful rhythm disorders in the MIT-BIH dataset and are widely used in the literature were considered. Information regarding these classes and the sample distribution within each class is presented in Table 1. The data used in this study were downloaded from the Physionet database [16].

Table 1. Distribution of ECG rhythm classes in the MIT-BIH Arrhythmia Dataset

Class	Description	Number of Sample
N	Normal Sinus Rhythm	90,000
S	Supraventricular Premature Beat	2,781
V	Premature Ventricular Contraction	7,236
F	Fusion of Ventricular and Normal Beat	803
Q	Unclassifiable Beats	9,905

B. Preprocessing and Beat Segmentation

The MIT-BIH Arrhythmia dataset's raw ECG recordings were preprocessed in this study to create pulse-level samples appropriate for the suggested Capsule Network model. In this case, every recording in the dataset was analyzed in conjunction with the annotation files that went with it. Figure 2 shows the pseudocode methods for importing the recordings into the system and processing them at the pulse level.

```
Algorithm 1: Pseudocode for ECG record reading and beat segmentation
Input: data_path, segment_length = 300
Output: segmented_beats, beat_labels
Initialize segmented_beats ← [ ];
Initialize beat_labels ← [ ];
Create record_list from all header files ending with ".hea";
foreach record_name in record_list do
    Read ECG record and corresponding annotation file;
    Extract signal, beat positions, and beat symbols;
    for each annotated beat do
        Define a fixed window of length segment_length around the beat position;
        if segment is within signal boundaries then
            Extract beat segment;
            Assign corresponding beat label;
            Store segment and label;
return segmented_beats, beat_labels;
```

Figure 2: **Pseudocode flow for ECG record reading and beat segmentation**

As shown in Figure 2, the record list was first created by scanning the header files in the data folder. Then, the ECG signal corresponding to each record and its annotation information were read. In this way, the continuous ECG recordings and the position and class information of each heartbeat were obtained simultaneously. The main operation in this stage was the segmentation of the ECG signal at the heartbeat level. By taking the annotation positions as reference points, a fixed-length window was defined for each beat, and each heartbeat was represented by 300 sample points. Thus, the continuous ECG recording was transformed into standard-length heartbeat segments suitable for model input. During segment extraction, only the beats that remained within the signal boundaries were included in the dataset. The main purpose of beat segmentation was not only to isolate individual heartbeats, but also to establish a consistent input format for the classification model. For this reason, each extracted heartbeat segment was added to the dataset together with its corresponding annotation label. In this way, the raw ECG data at the record level were converted into labeled and standardized samples at the heartbeat level. An example of a fragmented ECG signal is given in Figure 3.

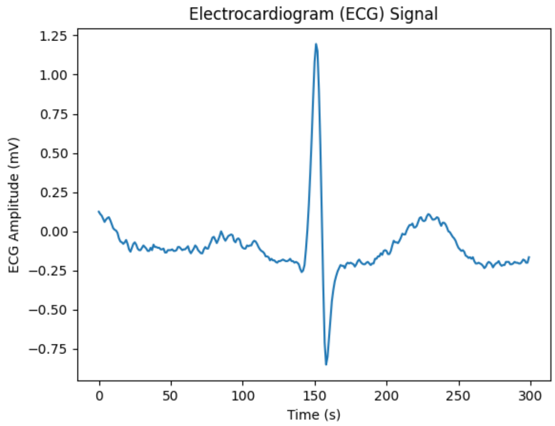

Figure 3: **Example of a fragmented ECG signal**

C. SMOTE-Based Data Balancing

One of the main challenges in the dataset used in this study is the imbalance among class distributions. As shown in the class distribution table, the N class contains a substantially larger number of samples than the remaining rhythm classes. Such an imbalanced structure may cause the classifier to become biased toward the majority class and may reduce its ability to learn the characteristic patterns of minority classes effectively. For this reason, a data balancing procedure was required before the model training stage. The Synthetic Minority Over-sampling Technique (SMOTE) was used to solve this issue. SMOTE is a data level balancing technique designed to reduce class imbalance in datasets with class imbalance. This technique achieves class imbalance by generating synthetic data instead of copying pre-existing data [17]. The risk of overfitting the classification algorithm is a problem to be solved in enhancements achieved with data enhancement techniques. It is suggested that the SMOTE technique can enrich data samples by using interpolations in the feature space, thereby reducing the risk of overfitting [17,18].

Algorithm-level approaches, data-level resampling approaches, and cost-sensitive learning approaches are the main categories into which imbalance managing techniques can be divided [17]. Among data-level techniques, random oversampling may continue to be unsuccessful since it merely replicates preexisting minority samples, while undersampling may eliminate important information from the majority class [17,19]. SMOTE, on the other hand, generates additional synthetic samples between nearby minority examples, making it a better choice when the goal is to maintain existing data while enhancing the representation of underrepresented classes [17,18]. Equation 1 gives the basic mechanism of the SMOTE technique. When SMOTE works, for each minority class, the nearest minority neighbors are determined and a synthetic sample is created along the line segment connecting the two samples [19].

$$S = x + u(x_R - x), \qquad \mathbf{0} \leq u \leq \mathbf{1} \tag{1}$$

When u is a random coefficient, x is the chosen minority class sample, and x_R is one of its closest minority neighbors. As a result, the generated sample is a new point in the feature space between two minority samples rather than a perfect replica of the original instance [16,18]. Therefore, in order to lessen the detrimental impact of class imbalance, SMOTE was applied to the input dataset as a preprocessing step prior to training the Capsule Network model. By decreasing the dominance of the majority class, it was intended to help the model acquire discriminative rhythm patterns and to boost the representation of data classes with a small number of samples.

D. Capsule Network-Based Classification Model

In this study, the Capsule Network (CapsNet) shown in Figure 4 was used as the classification model. Capsule networks are a deep learning architecture developed to reduce some of the limitations of conventional convolutional neural networks, particularly in representing spatial and hierarchical relationships [20]. The main difference in this structure is that, instead of producing scalar outputs as in classical neurons, groups of neurons called capsules produce vector outputs. A capsule output vector's direction expresses properties including position, scale, and orientation, while its length indicates the likelihood that the associated feature will be present [20, 21].

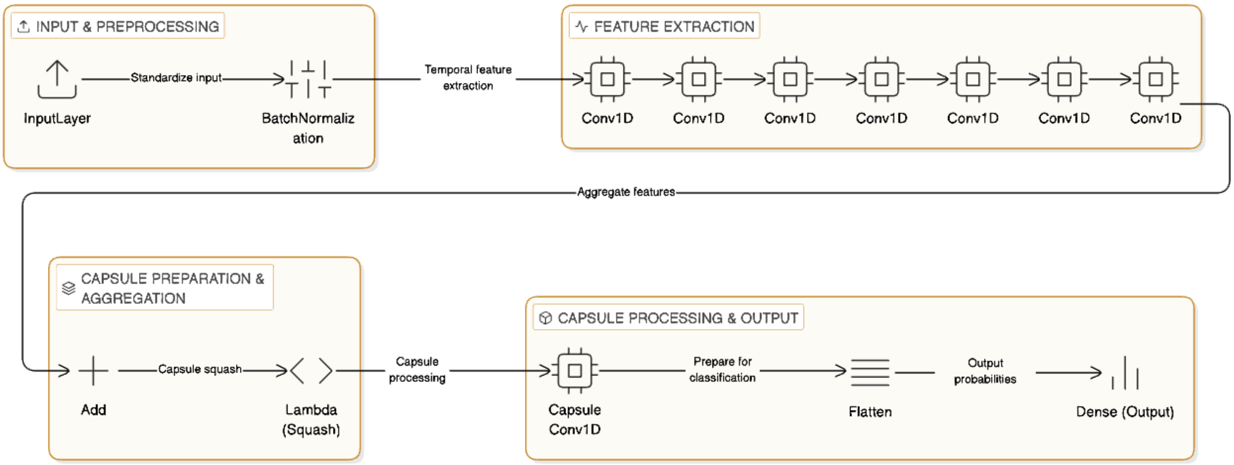

Figure 4: Proposed CapsNet Architecture

In this study, the first layer is the convolutional layer. The primary capsule layer receives the features extracted from the input data. The general expression for the convolutional layer's output is given in Equation 2.

$$z = W * x + b \tag{2}$$

Here, x denotes the input data, W denotes the convolution kernel, and b denotes the bias term. The activations obtained as a result of this operation are then transformed into capsule structures and converted into vector representations [20,21]. The most fundamental component of capsule networks is the dynamic routing mechanism, which enables information transfer from lower-level capsules to higher-level capsules. The aim of this mechanism is to iteratively determine which upper capsule should receive a greater contribution from the output of a lower capsule [20,21]. The dynamic routing process begins by initializing the coupling logits to zero, as shown in Equation 3:

$$b_{ij} = 0 \tag{3}$$

In this case, the initial coupling logit between the *i*-th lower-level capsule and the *j*-th upper-level capsule is denoted by b_{ij}. Next, the softmax function provided in Equation 4 is used to calculate the coupling coefficients, which establish the degree to which each lower-level capsule is coupled to upper-level capsules.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \tag{4}$$

The weight of the data transmitted from the *i*-th lower level and the k-th upper level of the capsule is denoted by c_{ij}. The sum of the connections from a lower level to all upper levels is 1 because the softmax function is used [20,21]. Equation 5 shows that the prediction vector is formed by transforming the output of each lower-level capsule with a weight matrix before it is projected into the higher-level capsule space.

$$\hat{u}_{j|i} = W_{ij} u_i \tag{5}$$

Here, u_i is the output vector of the lower-level capsule, and W_{ij} is the transformation matrix that goes from lower-level capsule i to upper-level capsule j. Equation 6 [20,21] shows how the coupling coefficients weight these prediction vectors to find the total input to the upper-level capsule:

$$s_j = \sum_i c_{ij} \, \hat{u}_{j|i} \tag{6}$$

This total vector is then passed through the squashing function to obtain the final capsule output, as shown in Equation 7:

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \tag{7}$$

The length of the capsule output vectors is compressed by this function to fall between 0 and 1. Consequently, short vectors are pushed closer to zero, while strong and meaningful representations are pushed closer to one. As a result, the length of the vector v can be interpreted as a probability value [20,21]. In the next routing stage, the coupling logits are updated using the formula from Equation 8.

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j \tag{8}$$

This update is performed according to the agreement between the prediction vector of the lower-level capsule and the actual output of the upper-level capsule. Here, the scalar product indicates how consistent the lower-level capsule's prediction is with the output of the upper-level capsule. As the agreement increases, the corresponding connection becomes stronger, so in subsequent iterations the information is transferred to the same upper-level capsule with a higher weight [20,21]. The main advantage of this structure is that the model can learn not only the presence of features, but also the relationships among these features [21].

E. Performance Metrics

In classification problems, the F1 score, recall, and accuracy metrics are used to analyze the results. The ratio of TP predictions to the sum of TP and FP predictions gives the accuracy metric. The equation required to calculate the accuracy metric is given in Equation 9 [22,23].

$$Precision = \frac{TP}{TP+FP} \tag{9}$$

where FP stands for false positives and TP for true positives. The recall rate is calculated from the ratio of the sum of true positive estimates, false negative estimates, and TP estimates. Equation 10 provides an evaluation of the model's capacity to find all pertinent cases [22,23].

$$Recall = \frac{TP}{TP+FN} \tag{10}$$

where the number of false negatives is denoted by FN. The F1 score is a metric that represents the balance between precision and recall. The F1 score plays an important role in examining unequal class distributions. Equation 11 is used to calculate the F1 score [22, 23].

$$F1\,Score = 2 \text{ x } \frac{Precision \text{ } x \text{ } Recall}{Precision+Recall} \tag{11}$$

Performance metrics play a significant role in the analysis of classification problems. Recall shows how well the system captures class differences, while the sensitivity metric shows how consistent the positively realized predictions are. The F1 score, derived from recall and sensitivity metrics, provides an objective evaluation of the system [22, 23].

III. RESULTS AND DISCUSSION

The effectiveness of the Capsule Network-based model for five-class ECG rhythm classification was assessed in this study using the MIT-BIH Arrhythmia Dataset. Five clinically significant rhythm classes—Q, F, S, V, and N—were used for the model's training and testing procedures. As a result, the study concentrated on precisely identifying various arrhythmia patterns within their respective classes in addition to assessing whether the beat was normal or pathological.

A. Experimental Results

In this study, parametric adjustments were made to the Capsule Network structure, but the basic architectural structure was preserved. The best performing parametric values were determined as a result of the experimental studies. In contrast to the last work, the model was intended to be trained for 150 epochs during the training phase. However, early halting and the best model checkpoint procedures were used to provide a more controlled training procedure because it was possible that such a lengthy training period could result in overfitting. In this case, the option that restores the best weights was enabled, the patience value was set to 10, and validation loss was chosen as the monitoring criterion. Therefore, the training process was automatically stopped and the model was kept with its optimal weights when no improvement in the validation loss was seen for ten epochs. This tactic caused training to end at the 132nd epoch rather than the intended 150 epochs. Additionally, the checkpointing method stored the best-performing model that was discovered during training independently. In addition to reducing the possibility of overfitting, this strategy made sure that the best model was used for the final assessments.

The confusion matrix for the training set is presented in Figure 5. Examination of the matrix shows that the correct classifications are largely concentrated along the diagonal. A total of 71,641 correct predictions were obtained for class N, 71,593 for class S, 72,105 for class V, 71,887 for class F, and 71,928 for class Q. This distribution indicates that the model learned all classes with a high level of accuracy on the training data. Nevertheless, a limited degree of confusion between classes was also observed. In particular, some samples belonging to class N were predicted as class S (437 samples), and some samples belonging to class F were confused with class V (187 samples), suggesting that the model may produce errors in patterns that are morphologically more similar to each other. However, these deviations remain low relative to the total number of samples.

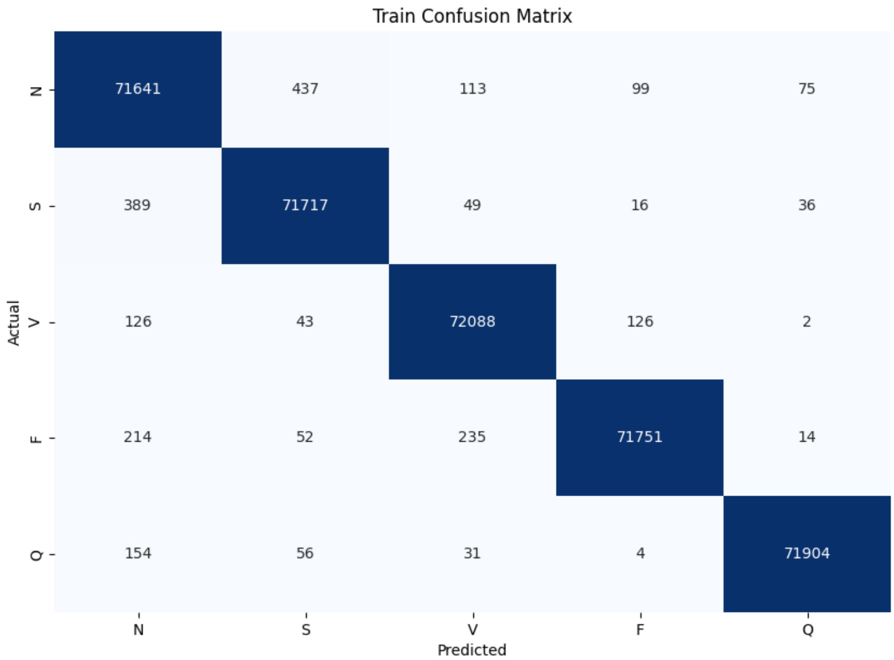

Figure 5: Confusion matrix of the Capsule Network model on the training set

Table 2 shows the measures of how well the training went. As the table shows, the precision, recall, and F1-score values for classes N and S are all 0.99. The F1-score for class V was 1.00, the recall was 1.00, and the precision was 0.99. For class F, the precision, recall, and F1-score were 1.00, 0.99, and 1.00, respectively. For class Q, all of these values were 1.00. The model learned all the classes equally well, not just the most important ones. This is shown by the training set's overall accuracy of 0.99 and the weighted-average and macro-average F1-scores being at the 0.99 level.

Table 2: Training performance metrics of the Capsule Network model

Class	Precision	Recall	F1-Score	Accuracy
N	0.99	0.99	0.99	0.99
S	0.99	0.99	0.99	0.99
V	0.99	1.00	1.00	0.99
F	1.00	0.99	1.00	0.99
Q	1.00	1.00	1.00	0.99

The confusion matrix for the test set is shown in Figure 6. The preservation of diagonal dominance in the test results indicates that the model did not merely memorize the training data, but also exhibited a similar classification behavior on unseen data. In the test set, 17,608 correct predictions were obtained for class N, 17,964 for class S, 17,885 for class V, 17,981 for class F, and 18,142 for class Q. When the distribution of errors across classes is examined, it is noteworthy that 33 samples belonging to class Q were classified as N, and 45 samples belonging to class F were predicted as V. This suggests that, particularly between certain classes, pattern similarities may create examples that are more difficult for the model to distinguish. Nevertheless, the proportion of these misclassifications within the total number of test samples remained limited.

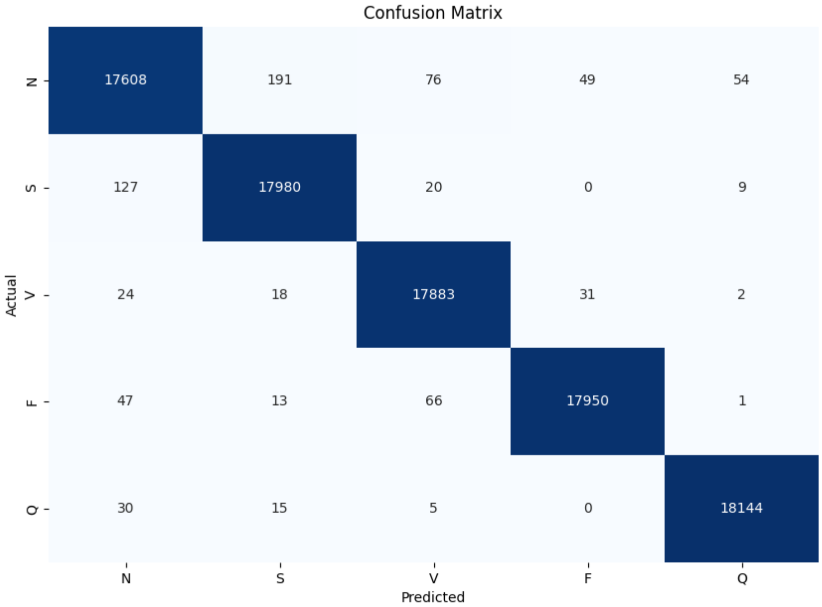

Figure 6: Confusion matrix of the Capsule Network model on the testing set

Performance metrics for each arrhythmia class are given in Table 3. Examining the metric values, the sensitivity value for classes N, S, and V of the proposed classification model was calculated as 0.99, while for classes F and Q it was 1.00. The sharpness value for classes S and F was calculated as 0.99, while for classes V and Q it was 1.00. The sharpness value for class N was found to be 0.98. The F1 score was obtained as 0.99 for classes S and V, 1.00 for classes F and Q, and 0.98 for class N. The analysis performed on the test set resulted in an accuracy of 99%. Examination of the confusion matrix indicates that the model performed well across all classes.

Table 3: Testing performance metrics of the Capsule Network model

Class	Precision	Recall	F1-Score	Accuracy
N	0.99	0.98	0.98	0.99
S	0.99	0.99	0.99	0.99
V	0.99	1.00	0.99	0.99
F	1.00	0.99	1.00	0.99
Q	1.00	1.00	1.00	0.99

When the training and test results obtained from the classification are examined, it is seen that similar performance metrics are obtained in both datasets. When the metrics and class distributions are examined, it is seen that the results are consistent for 5 different rhythms. The F1 score, sharpness, and precision values are obtained within the range of 0.98 to 1.00. The limited difference between the training and test results also suggests that the generalization capability of the model is at an acceptable level. Therefore, the findings indicate that the Capsule Network-based structure achieved acceptable results on the five-class ECG rhythm classification problem.

A. Discussion

In the study conducted, when the classification performance of the training and test datasets was examined, it was observed that 5 different classes of ECG rhythms were classified with an acceptable level of performance. The fact that the overall accuracy was maintained at the 0.99 level in both the training and test sets suggests that the model not only achieved a high level of fit on the training data but also maintained a similar classification behavior on the test data. The basic structure of the capsule network used in the study was preserved, with some limited modifications made to achieve high performance in classification. To prevent overlearning during the training process, loss of validation criteria were used as early stopping and monitoring criteria. The limit value here was chosen as 10. As a result of these parameter settings, the training process was completed in 132 iterations instead of the planned 150 iterations. This approach is considered to have contributed to a more stable training process by limiting the risk of overfitting. Analysis of the confusion matrices shows that the correct prediction classes are correctly classified in the diagonal plane in both the training and test classification results. This indicates that the proposed CapsNet architecture achieves an acceptable level of classification performance.

Nevertheless, a limited number of confusions between some classes were also observed. In the training set, misclassifications were particularly noticeable between the N-S and F-V class pairs, while in the test set, errors were again observed between F-V and Q-N. These observations may be related to the fact that these classes contain morphologically similar patterns. However, since the proportion of these confusions remained low within the total number of samples, it can be said that they did not substantially weaken the overall performance of the model.

As a result, it may be stated that this capsule network-based approach provides an applicable method for the five-class ECG rhythm classification problem. The obtained performance values indicate that the model is capable of producing balanced results in terms of both overall accuracy and class-based metrics. Although limited, the observed erroneous classifications between classes are thought to stem from morphological similarities in the data. Future studies could explore this as a potential area for addressing water problems.

IV. CONCLUSION

In this study, a Capsule Network-based method was developed for classifying five different ECG rhythm types using the MIT-BIH Arrhythmia Dataset. The study subjected the dataset to training and testing processes using rhythm classes Q, F, V, S, and N, which are considered significant. During the analysis process, ECG data were divided into predetermined lengths of samples corresponding to each heartbeat and then grouped. The SMOTE method was applied to reduce imbalance between classes. According to the results of the study, the proposed capsule network architecture achieved a satisfactory level of performance in both training and test data. The experimental study yielded an accuracy rate of 99%. Metrics derived from the confusion matrix used in the classification process—sharpness, precision, and F1 score—were obtained between 98% and 100%.These results demonstrate that the proposed capsule network architecture is highly effective in distinguishing ECG rhythms in five different classes. However, the very low error rates among some morphologically similar rhythm groups suggest that the model's method for distinguishing these classes needs to be examined in more detail. Accordingly, future studies could utilize more advanced feature extraction methods or explore different model architectures that can better distinguish similar ECG rhythms.

Interest Conflicts

The authors declares that there is no conflict of interest concerning the publishing of this paper.

Funding Statement

The authors received no financial support for the research, authorship, and/or publication of this article.

V. REFERENCES

[1] Ansari, Y., Mourad, O., Qaraqe, K., & Serpedin, E. (2023). Deep learning for ECG Arrhythmia detection and classification: an overview of progress for period 2017–2023. Frontiers in Physiology, 14, 1246746.

[2] Nahak, S., Pathak, A., & Saha, G. (2023). Evaluation of handcrafted features and learned representations for the classification of arrhythmia and congestive heart failure in ECG. Biomedical Signal Processing and Control, 79, 104230.)

[3] Cuenca, D. F. B., Serrezuela, R. R., & Gómez, A. E. R. (2025, September). Arrhythmia Classification from ECG Signals Using LSTM Neural Networks. In 2025 IEEE VIII Congreso Internacional en Inteligencia Ambiental, Ingenieria de Software y Salud Electronica y Movil (AmITIC) (pp. 1-7). IEEE.

[4] Li, Q., Liu, Y., Na, Z., Yuan, Y., & He, R. (2026). A novel ECG QRS complex detection algorithm based on dynamic Bayesian network. Artificial Intelligence in Medicine, 103370.

[5] Xiang, Y., Lin, Z., & Meng, J. (2018). Automatic QRS complex detection using two-level convolutional neural network. Biomedical engineering online, 17(1), 13.

[6] He, R., Liu, Y., Wang, K., Zhao, N., Yuan, Y., Li, Q., & Zhang, H. (2020). Automatic detection of QRS complexes using dual channels based on U-Net and bidirectional long short-term memory. IEEE journal of biomedical and health informatics, 25(4), 1052-1061.

[7] Haq, M. U., Sethi, M. A. J., & Rehman, A. U. (2023). Capsule network with its limitation, modification, and applications—A survey. Machine Learning and Knowledge Extraction, 5(3), 891-921.

[8] Ullah, H., Bin Heyat, M. B., Akhtar, F., Sumbul, Muaad, A. Y., Islam, M. S., ... & Lai, D. (2022). An End-to-End Cardiac Arrhythmia Recognition Method with an Effective DenseNet Model on Imbalanced Datasets Using ECG Signal. Computational Intelligence and Neuroscience, 2022(1), 9475162.

[9] Mangaraj, S., Mahapatra, K., & Ari, S. (2025). Cardiac arrhythmia classification system: An optimized HLS-based hardware implementation on PYNQ platform. Microprocessors and Microsystems, 105225.

[10] Daydulo, Y. D., Thamineni, B. L., & Dawud, A. A. (2023). Cardiac arrhythmia detection using deep learning approach and time frequency representation of ECG signals. BMC Medical Informatics and Decision Making, 23(1), 232.

[11] Asif, M. S., Faisal, M. S., Dar, M. N., Hamdi, M., Elmannai, H., Rizwan, A., & Abbas, M. (2023). Hybrid deep learning and discrete wavelet transform-based ECG biometric recognition for arrhythmic patients and healthy controls. Sensors, 23(10), 4635.

[12] Moody, G. B., & Mark, R. G. (2001). The impact of the MIT-BIH arrhythmia database. IEEE engineering in medicine and biology magazine, 20(3), 45-50.

[13] Zhou, Y., Tian, J., & Kang, K. (2026). Multimodal Signal Processing of ECG Signals with Time-Frequency Representations for Arrhythmia Classification. Computer Modeling in Engineering & Sciences, 146(2).

[14] Hammad, M., Kandala, R. N., Abdelatey, A., Abdar, M., Zomorodi-Moghadam, M., San Tan, R., ... & Pławiak, P. (2021). Automated detection of shockable ECG signals: A review. Information Sciences, 571, 580-604.

[15] Mohammed, A. A., Al-Sulami, Z. A., Abd Zaid, M. M., Abduljabbar, Z. A., Aldarbandee, M., Jaber, H., ... & Neamah, H. A. (2026). Enhanced Arrhythmia Diagnosis Using a Hybrid Deep Learning Model: A CNN-LSTM-GRU Approach on ECG Data. Biosensors and Bioelectronics: X, 100767.

[16] Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation, 101(23), e215–e220. https://doi.org/10.1161/01.CIR.101.23.e215

[17] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[18] Pradipta, G. A., Wardoyo, R., Musdholifah, A., Sanjaya, I. N. H., & Ismail, M. (2021, November). SMOTE for handling imbalanced data problem: A review. In 2021 sixth international conference on informatics and computing (ICIC) (pp. 1-8). IEEE.

[19] Fransico J Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. BMC bioinformatics, 14(1), 106.

[20] Dombetzki, L. A. (2018). An overview over capsule networks. Network architectures and services, 10.

[21] Marchisio, A., Bussolino, B., Salvati, E., Martina, M., Masera, G., & Shafique, M. (2022, August). Enabling capsule networks at the edge through approximate softmax and squash operations. In Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design (pp. 1-6).

[22] Qian, C., Sun, Z., Wang, C., Tian, E., & Hu, X. (2026). Self-Knowledge Distillation from Attention-Enhanced Multi-branch Residual Network for 12-Lead Electrocardiogram Arrhythmia Classification. Digital Signal Processing, 106043.

[23] Yacouby, R., & Axman, D. (2020, November). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In Proceedings of the first workshop on evaluation and comparison of NLP systems (pp. 79-91).