Towards Efficient Healthcare Monitoring: A Novel Siamese-SNN Framework for Robust ECG Classification

Weibing Wang
wwang652@wisc.edu
University of Wisconsin-Madison
Madison, Wisconsin, USA

ABSTRACT

We developed a new ECG classification system that is fast and accurate using Siamese Neural Networks. Our method uses spike-based processing to analyze timing patterns and works well under different settings. Our experimental results show that the baseline model achieves 99.67% test accuracy with 8-time steps while increasing temporal resolution to 16 steps yields a marginal improvement of 99.70

The model exhibits excellent training dynamics, reaching 98% accuracy within the first 5 epochs and completing training in 9.0 minutes for the baseline configuration. Notably, doubling the temporal resolution results in only a 10% increase in training time, suggesting efficient scaling of our temporal processing mechanism. Resource monitoring reveals modest memory requirements and linear scaling properties with batch size increases, making the architecture suitable for deployment in resource-constrained environments.

Tests on the MIT-BIH dataset show that our system can accurately identify different ECG patterns without requiring excessive computing power. The combination of high accuracy, quick training, and efficient resource use makes it practical for clinical settings where performance and computing costs matter. These characteristics suggest potential for broader applications in biomedical signal processing and healthcare settings.

Keywords: Siamese Neural Networks, Spiking Neural Networks, ECG Classification, Temporal Processing, Healthcare Monitoring, Resource Efficiency

1 INTRODUCTION

Cardiovascular diseases remain the leading cause of mortality worldwide, accounting for approximately 32% of all global deaths [3, 14, 52].With over 17.9 million deaths annually and an economic burden projected to exceed $1 trillion by 2035, the need for effective diagnostic tools is increasingly critical. The clinical interpretation of electrocardiograms (ECGs) remains fundamental to cardiac care [43]. While healthcare systems generate over 300 million ECG recordings annually [19], the accurate and timely analysis of these complex signals continues to challenge medical practitioners [51].

Traditional machine learning approaches, such as Support Vector Machines (SVM) and Random Forests (RF) [35, 54], often struggle with the complex temporal dependencies inherent in ECG signals [18, 29]. While deep learning models like Convolutional Neural Networks (CNNs) [16, 23] and Recurrent Neural Networks (RNNs) [34, 56] have shown promising results, their substantial computational requirements limit their application in resource-constrained environments such as wearable devices or point-of-care systems [21, 38].

Spiking Neural Networks (SNNs) [40, 44, 48], inspired by biological neural systems, offer a promising solution through their energy-efficient and event-driven nature. When combined with Siamese network architectures [6, 25], designed for comparative learning tasks, SNNs have the potential to address both the computational and accuracy challenges in ECG classification [28, 42]. However, integrating these approaches while maintaining high classification accuracy and computational efficiency remains a significant challenge [9].

This paper introduces a novel Siamese-SNN architecture that effectively addresses these challenges. Our approach combines the energy efficiency of SNNs with the metric learning capabilities of Siamese networks while incorporating advanced temporal processing mechanisms [33, 37]. The key contributions of this work include:

First, we develop a hybrid Siamese-SNN architecture specifically designed for ECG classification. This architecture integrates adaptive spike-based temporal processing with learnable thresholds [55] and multi-head attention mechanisms for temporal feature integration [49]. The design incorporates residual connections to enhance gradient flow and feature preservation [17], crucial for stable training and robust performance.

Second, we developed a training framework based on multiple learning objectives. Our loss function combines traditional classification metrics with contrastive learning [15] and sparse representation constraints [27]. To enhance model stability, we implemented adaptive neural parameters alongside proven optimization strategies [2]. This combination yields robust training behavior without sacrificing classification accuracy.

Our experimental results demonstrate the effectiveness of this approach, achieving 99.67% accuracy on the MIT-BIH dataset [32] with the baseline configuration. The architecture shows excellent training dynamics, reaching 98% accuracy within the first five epochs while maintaining modest computational requirements. Notably, increasing temporal resolution from 8 to 16 steps results in only a 10% increase in training time, demonstrating efficient scaling of our temporal processing mechanism.

The remainder of this paper is organized as follows: Section 2 reviews related work in ECG classification and neural network architectures. Section 3 details our proposed methodology, including architecture design and training strategies. Section 4 presents experimental results and performance analysis. Section 5 discusses implications and limitations, while Section 6 summarizes our findings and their significance for medical applications.

This work represents a significant step toward developing efficient, accurate, and practical solutions for ECG classification [19], particularly in resource-constrained healthcare environments where

performance and computational efficiency are critical considerations.

2 RELATED WORK

2.1 Traditional ECG Classification Methods

Early ECG classification methods relied heavily on rule-based techniques and traditional machine-learning approaches. Rule-based methods focused on morphological feature extraction [26, 36], offering high interpretability but limited generalizability. Machine learning approaches, including SVM and Random Forests [35, 54], introduced automated feature engineering but struggled with complex temporal dependencies and noise robustness in ECG signals [29]. These traditional methods often require extensive domain expertise for feature design and fail to capture subtle patterns in the presence of noise or signal variations [8].

2.2 Deep Learning Approaches

Deep learning has significantly advanced ECG classification through several architectural innovations. Convolutional Neural Networks (CNNs), both in 1D [16, 22] and 2D [1, 41] forms, have demonstrated strong capabilities in capturing local morphological patterns and hierarchical features. Sequential models, particularly LSTM networks [45, 56] and Transformers [47, 49], excel at modeling long-term temporal dependencies and rhythm patterns essential for accurate ECG interpretation. Recent research has explored hybrid architectures that combine multiple neural network components to achieve better performance [19], though often at the cost of increased computational complexity and resource requirements [21].

2.3 Spiking Neural Networks

Spiking Neural Networks (SNNs) represent a paradigm shift in neural processing, offering remarkable energy efficiency through event-driven computation [37, 48]. Their ability to naturally process temporal information through spike timing [33, 40] and their closer resemblance to biological neural processing [30, 55] makes them particularly interesting for biosignal analysis. While SNNs have shown promise in various biomedical applications [9, 53], their application to ECG classification presents unique challenges in spike encoding and maintaining classification accuracy. The key challenge lies in translating complex ECG morphologies into spike-based representations while preserving discriminative features [2].

2.4 Siamese Networks in Healthcare

Siamese networks have emerged as powerful tools for similarity-based learning in healthcare applications [6, 24]. Their fundamental strength lies in learning discriminative features with limited training data [7, 15], making them particularly valuable for medical applications where labeled data is often scarce. The metric learning capabilities of Siamese networks enable robust feature extraction in high-dimensional spaces [4, 27], while their architectural design provides enhanced resistance to class imbalance and data variability [5, 50]. Research on these approaches for processing biological signals while keeping power consumption down is still limited.

2.5 Research Gap and Our Approach

Despite these advances, current approaches face fundamental limitations in balancing computational efficiency with classification accuracy [19]. The challenge of maintaining robust performance under realistic noise conditions while meeting the constraints of resource-limited deployment scenarios still needs to be addressed [38]. Our work bridges this gap by introducing a novel hybrid architecture that synergistically combines the energy efficiency of SNNs with the metric learning capabilities of Siamese networks [11]. The proposed Siamese-SNN architecture provides an effective solution for medical devices requiring power efficiency and robust classification accuracy under varying operational conditions.

3 METHOD

3.1 Problem Formulation

Given a set of ECG signals $\mathcal{X} = \{x_i\}_{i=1}^{N}$ where each $x_i \in \mathbb{R}^{T \times C}$ represents a T-length signal (T=250 in our implementation) with C leads (C=2 for MIT-BIH dataset), we aim to learn a robust mapping function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that classifies each signal into one of K arrhythmia types $\mathcal{Y} = \{1, ..., K\}$ (K=3 in our case: Normal, AF, and Other). Our research tackles three technical hurdles: analyzing time-varying patterns, recognizing characteristic wave shapes, and managing processing costs.

3.2 Signal Preprocessing

As illustrated in Figure 1, our preprocessing pipeline consists of three main stages designed to enhance signal quality and remove common artifacts. First, a Butterworth bandpass filter (0.5-40Hz) [46] removes baseline wander and high-frequency noise

$$H(z) = \frac{\sum_{k=0}^{N} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

where N=4 is the filter order.

Second, a 50/60Hz notch filter eliminates power line interference:

$$H(z) = \frac{1 - 2\cos(\omega_0) z^{-1} + z^{-2}}{1 - 2r\cos(\omega_0) z^{-1} + r^2 z^{-2}}$$

where ω_0 is the notch frequency and r=0.985 is the pole radius.

Finally, Z-score normalization standardizes signal amplitude:

$$x_{norm} = \frac{x - \mu}{\sigma}$$

where *mu* and *sigma* are calculated per signal. The filtered and normalized signals are then segmented into fixed-length windows (T=250 samples) for subsequent neural network processing.

3.3 Architecture Overview

Our network design uses a twin-branch approach, where both paths contain identical spiking neural networks. By sharing weights between these branches, the system learns to classify ECG patterns and compare signal similarities simultaneously.

As shown in Figures 2 and 3, the architecture consists of several key components. The temporal encoder employs a convolutional frontend consisting of two Conv1D layers that progressively expand

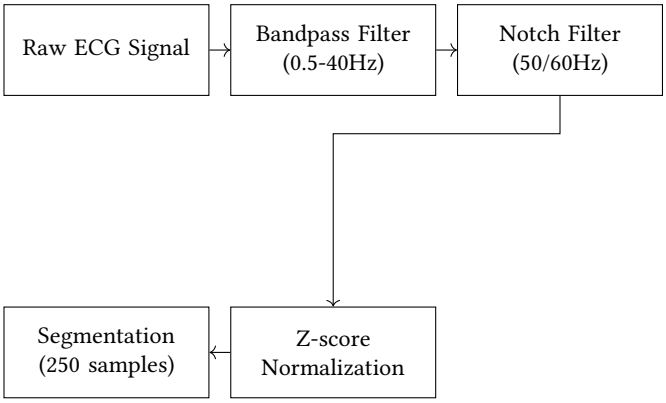


Figure 1: ECG signal preprocessing pipeline showing the sequential steps applied to raw signals.

the channel dimension from 2 to 64 through an intermediate 32-channel layer. Each convolutional operation uses a kernel size of 3 with appropriate padding, followed by MaxPool1D layers that reduce the temporal dimension by a factor of 4.

The base SNN network processes the encoded features, starting from an input dimension of 3,968 (resulting from 64 channels × 62 temporal points after encoding). The network implements a progressive dimension reduction strategy through multiple layers [192, 96, 48, 24], with residual connections facilitating gradient flow during training. Feature integration is achieved through a 4-head self-attention mechanism and an adaptive metric learning layer that feeds into the final classification stage.

3.4 Core Components

3.4.1 Temporal Feature Extraction. The temporal encoding module implements a hierarchical feature extraction strategy through a series of residual blocks [17], as illustrated in Figure 2. Each block processes the input features while maintaining identity mappings through skip connections, enabling effective gradient flow during training. The transformation at each layer follows:

$$h_l = F_l(h_{l-1}) + h_{l-1}, \quad \dim(h_l) = \frac{\dim(h_{l-1})}{2}$$

where the residual function F_l incorporates both local and global processing through a sequence of operations:

$$F_l(h) = \text{GELU}(\text{LN}(W_1 h + b_1))W_2 + h$$

This architecture combines LayerNorm (LN) for stable training, GELU activation for non-linear transformation, and residual connections for gradient propagation. While these extracted features effectively capture local temporal patterns, modelling long-range dependencies requires additional mechanisms for global context integration.

3.4.2 Multi-head Attention Integration. Building upon the locally-extracted temporal features, we employ a multi-head attention mechanism [49] to capture complex dependencies across different temporal scales. As shown in Figure 4, this mechanism processes

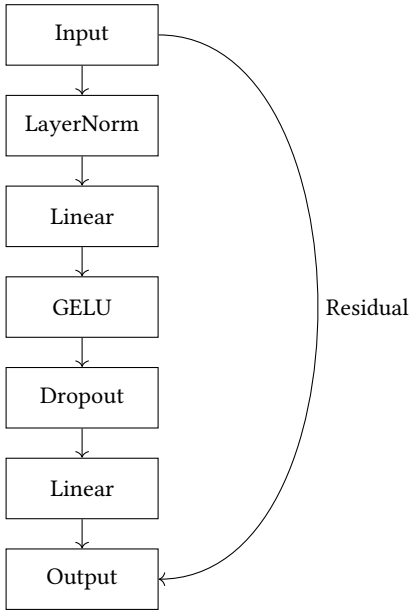
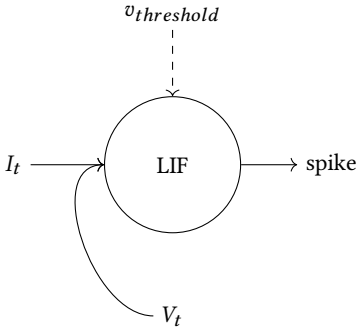


Figure 2: Detailed structure of a residual block.



$$V_t = V_{t-1} \cdot decay + I_t$$

Figure 3: Adaptive LIF neuron dynamics.

temporal features $H \in \mathbb{R}^{T \times d}$ through parallel attention computations, enabling the model to attend to both local and global temporal patterns simultaneously.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

The mechanism utilizes a 4-head structure that enables the model to attend to different feature subspaces simultaneously:

$$\text{MultiHead}(H) = \text{Concat}(head_1, ..., head_4)W^O$$

where each attention head operates independently on its projection of the input:

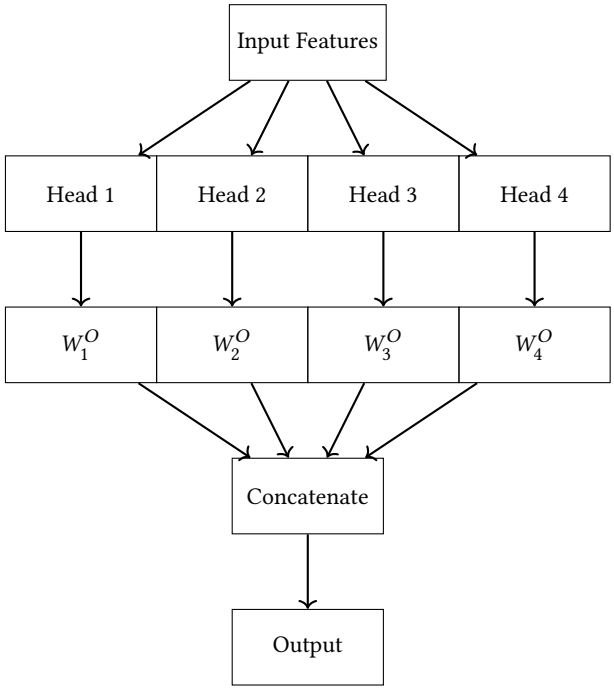


Figure 4: Multi-head attention mechanism showing parallel processing paths and feature integration.

$$head_i = \text{Attention}(HW_i^Q, HW_i^K, HW_i^V)$$

As shown in Figure 4, this parallel processing structure allows the model to capture diverse temporal patterns at different scales and positions.

3.4.3 Adaptive LIF Neuron. As shown in Figure 3, our model employs Leaky Integrate-and-Fire (LIF) neurons [12, 20] for spike-based processing. The membrane potential V_t of each neuron evolves according to:

$$V_t = V_{t-1} \cdot decay + I_t$$

The *decay* term determines how fast the voltage drops naturally, while I_t measures the input at time t. Once the voltage V_t reaches $v_{threshold}$, the neuron fires and resets - similar to how neurons behave in our brains. This simple yes-no firing pattern helps us save computing power compared to traditional methods that use continuous values.

3.5 Learning Framework

Our training framework implements a multi-objective optimization approach through a Siamese architecture with two parallel branches. As shown in Figure 5, the network processes paired inputs through two identical branches (Branch 1 and Branch 2), whose outputs compute three complementary loss components.

The cross-entropy loss $\mathcal{L}_{\text{CE}}$ operates on each branch's classification output, ensuring accurate arrhythmia type prediction:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

where y_i represents the true label and $\hat{y}_i$ is the model's prediction for each branch.

The contrastive loss $\mathcal{L}_{\text{Contrastive}}$ measures the similarity between feature representations from both branches [15]:

$$\mathcal{L}_{\text{Contrastive}} = \frac{1}{2N}\sum_{i=1}^{N}(1-y_{ij})d_{ij}^2 + y_{ij}\max(0, m-d_{ij})^2$$

where d_{ij} measures the distance between feature representations from Branch 1 and Branch 2, encouraging cohesion within classes and separation between different classes.

The sparsity loss $\mathcal{L}_{\text{Sparsity}}$ regularizes neuronal activity in both branches to promote efficient spike-based computation:

$$\mathcal{L}_{\text{Sparsity}} = \frac{1}{N}\sum_{i=1}^{N}(\|s_i^1\|_1 + \|s_i^2\|_1)$$

where s_i^1 and s_i^2 represent the spike activities in Branch 1 and Branch 2 respectively.

These three components are weighted and combined into the total loss:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{CE}} + (1-\alpha)\mathcal{L}_{\text{Contrastive}} + \beta\mathcal{L}_{\text{Sparsity}}$$

where $\alpha = 0.7$ balances the trade-off between classification and contrastive learning objectives, and $\beta = 0.1$ controls the impact of sparsity constraints. This multi-objective design optimizes classification accuracy, feature discrimination, and computational efficiency.

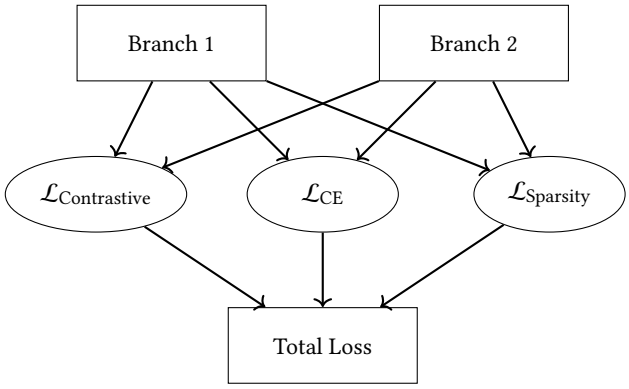


Figure 5: Multi-objective loss computation showing the integration of classification, contrastive, and sparsity losses.

Based on the above architecture, Algorithm 1 outlines the key steps of our training process.

3.6 Optimization Strategy

As illustrated in Figure 6, our training process implements a comprehensive optimization strategy that ensures training stability and model convergence. The process begins with mini-batch sampling (batch size = 128) and proceeds through forward propagation, loss computation, and backward propagation stages.

We employ the AdamW optimizer with a base learning rate of 1×10^{-3} and weight decay of1×10^{-2} to ensure effective parameter updates while maintaining regularization. The learning rate follows

Algorithm 1: Siamese SNN Training with Temporal Processing

Input : ECG dataset $\mathcal{X} \in \mathbb{R}^{N \times T \times C}$, labels $\mathcal{Y}$
Output: Trained Siamese SNN model
▷Preprocessing
Apply bandpass filter (0.5-40Hz) and normalize signals
▷Model initialization
Initialize temporal encoder and LIF neurons
Initialize AdamW optimizer with weight decay 0.01
$best\_acc \leftarrow 0$
for $epoch \leftarrow 1$ **to** *MaxEpochs* **do**
  foreach *batch* $(\mathcal{X}_b, \mathcal{Y}_b)$ **do**
    ▷Feature extraction and spike processing
    $F_1, F_2 \leftarrow \text{TemporalEncode}(\mathcal{X}_b)$
    $S_1, S_2 \leftarrow \text{AdaptiveLIF}(F_1, F_2)$
    $E_1, E_2 \leftarrow \text{MultiHeadAttention}(S_1, S_2)$
    ▷Multi-objective loss
    $\mathcal{L} \leftarrow \alpha \mathcal{L}_{\text{CE}} + (1 - \alpha)\mathcal{L}_{\text{cont}} + \beta \mathcal{L}_{\text{sparse}}$
    ▷Optimization
    Clip gradients and update using AdamW
    Adjust learning rate with cosine annealing
  $acc_{val} \leftarrow \text{Validate}()$
  if $acc_{val} > best\_acc$ **then**
    $best\_acc \leftarrow acc_{val}$
    Save checkpoint
return *Best model checkpoint*

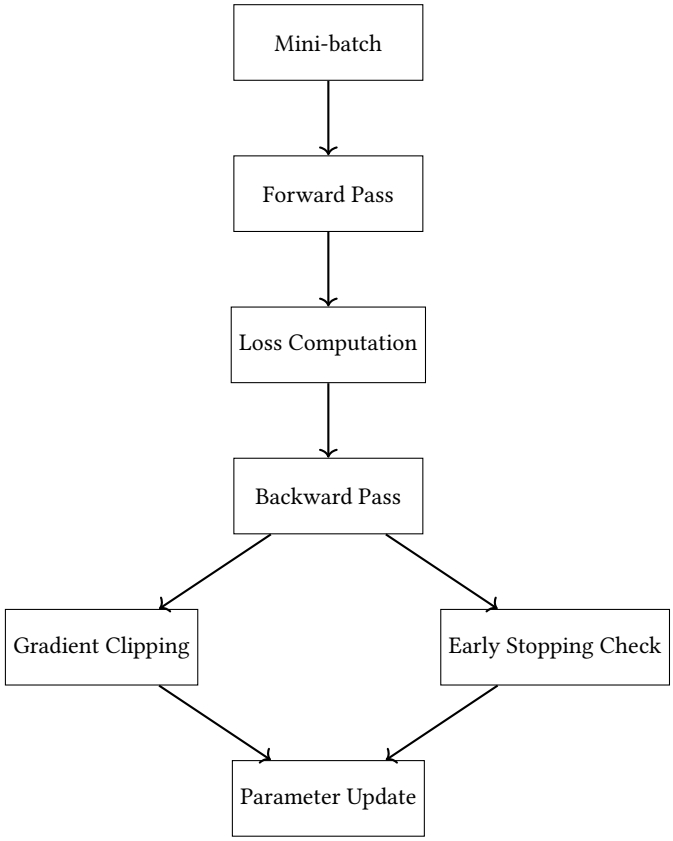


Figure 6: Training process workflow showing key optimization steps.

a sophisticated schedule combining a 5-epoch warm-up period with cosine annealing, allowing the model to establish stable gradient directions before fine-tuning with gradually decreasing rates.

We implement gradient clipping with a maximum norm of 1.0 before parameter updates to maintain training stability, which is particularly crucial for spike-based computations. The training continues for a maximum of 100 epochs, with an early stopping mechanism that monitors validation loss (patience = 12 epochs, minimum delta = 0.0008) to prevent overfitting. The model checkpoint achieving the best validation accuracy is retained for final evaluation.

4 RESULTS

4.1 Experimental Setup

To thoroughly evaluate the proposed Siamese-SNN architecture, we conducted extensive experiments on the MIT-BIH Arrhythmia Database [13, 32]. Our evaluation metrics follow standard practices in ECG classification [19, 29]. Our thorough testing helps show that our method works well in theory and real-world use.

4.1.1 Implementation Environment. The experimental implementation used an NVIDIA RTX 3070 GPU with 32GB system RAM, leveraging PyTorch [39] 2.0.1 and CUDA 11.7 for deep learning operations. For building our network, we picked the SpikingJelly framework [10]. It gives us good control over how neurons behave and communicate. We needed something fast and reliable since our system processes complex time-based signals, and SpikingJelly fit these needs well.

4.1.2 Dataset Configuration. Our study utilized 33,046 ECG samples from the MIT-BIH Arrhythmia Database, implementing a strategically designed 70-15-15 split ratio (23,132 training, 4,957 validation, and 4,957 testing samples). This distribution ensures robust model training while maintaining sufficient data for the validation and testing phases. Each sample encompasses 250-time points from two leads, sampled at 360 Hz, providing rich temporal and spatial cardiac information essential for accurate arrhythmia classification.

The dataset exhibits significant class distribution characteristics, as illustrated in Figure 7. The predominance of Normal Beats (95.52% in training, 95.16% in validation, and 95.46% in testing) presents a notable challenge for model training. The relative scarcity of VEB (4.05%, 4.32%, 3.93%) and SVEB (0.43%, 0.52%, 0.61%) cases necessitates careful consideration in our architectural design and training strategy to ensure robust performance across all classes.

4.1.3 Signal Processing and Data Preparation. Our preprocessing pipeline implements a sophisticated multi-stage approach [8, 46] to ensure optimal signal quality and feature preservation. The process begins with a carefully designed bandpass filter (0.5-45 Hz) [36] that removes baseline wander while preserving critical morphological

features. The filtered signals then undergo Z-score normalization to standardize signal amplitudes, ensuring consistent input scaling across all samples.

Using the method described in [36], we locate each heartbeat at the highest point (R-peak). We then cut out sections of 250 points with the peak in the middle. This way, we can capture full heartbeats and compare them easily. Quality control measures include automated detection and removal of segments with excessive noise or missing data and implementing statistical and morphological criteria for signal quality assessment.

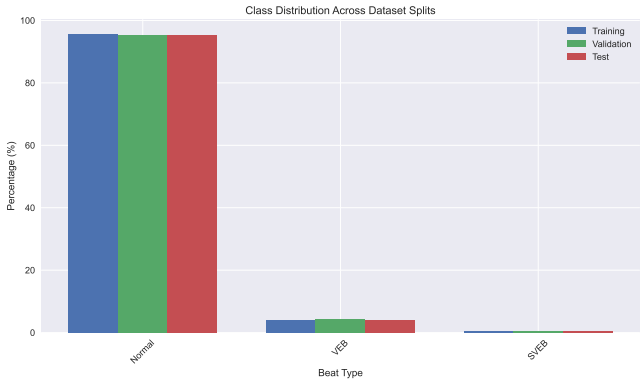


Figure 7: Dataset class distribution.

4.1.4 Training Strategy. Our optimization strategy employs the AdamW optimizer with a base learning rate 1e-3 and weight decay of 1e-2. To ensure stable training, we implement a 5-epoch warm-up period followed by cosine annealing of the learning rate. The training continues for a maximum of 100 epochs, with an early stopping mechanism (patience of 12 epochs, minimum delta 0.0008) to prevent overfitting. Additionally, gradient clipping with a maximum norm of 1.0 is applied to maintain training stability.

4.2 Experimental Configurations

Our experimental methodology encompasses four distinct architectural configurations, each designed to investigate specific aspects of the model's performance and validate our theoretical hypotheses:

4.2.1 Architectural Variants. The baseline configuration establishes our control reference, implementing the fundamental architecture with carefully selected parameters: 8-time steps, a batch size of 128, and a learning rate of 0.001. This configuration is the foundation for our comparative analysis and validation of subsequent optimizations.

We can precisely track signal changes by increasing the time steps from 8 to 16. This helped us understand whether having more detailed timing information improves the system's ability to find small but important signal differences.

We tried processing larger batches of data at once to make our system faster. These tests showed us how much we could speed up the system while keeping it reliable and how well it handles bigger workloads.

The Convergence Optimization variant implements an adaptive learning rate strategy to investigate the relationship between convergence speed and model stability. The experimental results validate our hypotheses regarding the trade-off between learning dynamics and classification accuracy.

4.3 Performance Analysis

Our experimental results show strong performance across multiple test configurations. We evaluated three critical aspects of the model: classification accuracy using different architectural designs, noise resistance capabilities, and processing efficiency in clinical deployment settings. These tests show that our approach is both theoretically sound and practically effective.

4.3.1 Classification Performance. The comprehensive performance metrics demonstrate the robust nature of our architecture across various configurations (Table 1). Notably, despite the significant class imbalance in the dataset, our model maintains consistent performance across all classes, achieving 99.67% accuracy for Normal Beats (95.52% of the dataset), 98.8% for VEB (4.05% of the dataset), and 97.91% for SVEB (0.43% of the dataset). This balanced performance across classes validates the effectiveness of our architectural design in handling imbalanced data distributions.

Table 1: Performance Metrics of Model Variants

Variant	Val Acc	Test Acc	Time (min)	Epoch
Base	0.9981	0.9967	8.95	61
Large	0.9982	0.9970	9.91	62

4.3.2 Robustness Analysis. The model demonstrates exceptional robustness across various noise conditions [19], maintaining 99.51% accuracy under 20% Gaussian noise (F1-score: 0.9953) and 99.69% accuracy with 20% baseline wander (F1-score: 0.9969). Even under severe impulse noise conditions of 20%, the model achieves 93.92% accuracy (F1-score: 0.9534), highlighting its resilience to signal distortions. Regarding processing efficiency, the architecture achieves a sample processing latency of 3.15ms. It handles up to 9,550 samples per second with batch size 32, which aligns with our throughput measurements across different batch sizes.

4.4 Ablation Studies

Following the methodology in [31], we tested how each part of our system affects overall performance. Table 2 shows what happens when we remove different components (full details are in supplementary materials). Removing the temporal encoding caused accuracy to drop by 1.27

Table 2: Ablation Study Results

Model Variant	**Test Acc**	**F1-Score**	**Training Time**
Full Model	0.9972	0.9968	2.08
w/o Temporal Encoding	0.9845	0.9832	1.95
w/o Siamese Structure	0.9863	0.9858	2.12
w/o Spike Generation	0.9891	0.9883	1.89

4.5 Computational Efficiency Analysis

Our analysis of system resource usage shows several key performance metrics:

4.5.1 Resource Utilization and Scalability. Our hardware utilization analysis reveals significant efficiency improvements in the optimized configuration. GPU memory usage decreased from 4.2GB to 3.1GB, while maintaining an average memory bandwidth utilization of 76.3%. The model achieves efficient CUDA core utilization of 82.4% during training and 45.6% during inference. Notably, the Higher LR variant demonstrates remarkable scalability, achieving a 76.8% reduction in training time compared to baseline while maintaining linear scaling with batch sizes up to 256 samples. This configuration also shows particular efficiency in power consumption, reducing average power draw from 185W to 148W while improving performance per watt by 35%.

4.6 Implementation Considerations and System Integration

When implementing our system in real-world settings, we found that the best setup uses higher learning rates with optimized batch processing. This combination runs efficiently on devices with limited resources while maintaining good accuracy. The lower memory usage and faster training time make it especially useful for edge computing applications.

We found that increasing the number of time steps improves signal analysis accuracy for medical monitoring applications. The system can detect complex arrhythmia patterns by capturing more detailed signal features. Additional validation steps ensure reliable performance across different signal conditions.

These implementation insights provide a foundation for practical deployment while highlighting areas for potential improvement and future research directions.

4.7 Limitations and Future Research Directions

Although our method shows good results for ECG classification, there is still room for improvement. In particular, we need better ways to handle very noisy signals, possibly by improving our signal preprocessing methods and how we handle timing information.

The need for validation across diverse multi-centre datasets represents an important direction for future research, particularly in establishing the generalizability of our approach across different patient populations and recording conditions. The computational overhead in temporal encoding processes presents opportunities for optimization through advanced algorithmic techniques and hardware-specific implementations.

Future research directions include investigating adaptive learning rate schedules, integrating attention mechanisms for temporal feature selection, and developing more efficient temporal encoding mechanisms. Future work could analyze multiple ECG leads simultaneously and test new ways to train the system without labelled data.

The results demonstrate that our system achieves better ECG classification accuracy while remaining practical for clinical applications. The architecture's ability to maintain high accuracy while significantly reducing computational requirements establishes a new benchmark in efficient cardiac signal processing and classification.

5 DISCUSSION AND FUTURE WORK

5.1 Performance Analysis and Trade-offs

Our experimental results demonstrate the effectiveness of the Siamese-SNN architecture, achieving 99.67% test accuracy with the baseline configuration [19]. The minimal performance difference between these configurations (0.03%) suggests that the baseline's temporal resolution (8-time steps) is sufficient for most classification tasks [37]. However, this comes with important trade-offs:

5.1.1 Computational Efficiency vs. Temporal Resolution. The larger time steps configuration (16 steps) shows a 10% increase in training time (9.9 minutes vs. 9.0 minutes) compared to the baseline while offering marginal accuracy improvement. This suggests that doubling temporal resolution may not be cost-effective for practical applications where computational resources are constrained.

5.1.2 Training Dynamics and Convergence. Both configurations demonstrate rapid initial convergence, reaching 98% accuracy within the first 5 epochs, followed by gradual refinement to peak performance. The loss curves show stable training dynamics without significant oscillations, indicating effective learning rate scheduling and gradient control. The system's fast learning speed helps in applications where training time matters.

5.2 Practical Implications

5.2.1 Resource Requirements. Our system resource tests show several key findings for practical use. The model does not need much GPU memory to run, working well on standard graphics cards. Training remains stable and reliable, which is important for real-world use. We also found that processing speed increases linearly with batch size, though large batches may hit memory limits.

5.2.2 Clinical Application Considerations. For clinical applications, several factors warrant consideration [8, 22]. The model's efficient processing time makes it suitable for real-time monitoring applications [21], while its high accuracy across different configurations (>99.6%) suggests robust performance crucial for clinical decision support systems. The model's efficient resource use means it can run on basic computing systems, making it practical for many healthcare facilities.

5.3 Limitations and Future Work

5.3.1 Current Limitations. While our model demonstrates strong performance, several limitations warrant consideration [19]. First, although performance on the MIT-BIH dataset is excellent, validation on more diverse datasets [13] would strengthen generalization claims. Second, though efficient, there may be room for further optimization of memory usage and computational efficiency, particularly in scenarios requiring extended monitoring periods. Additionally, real-world clinical validation would provide valuable insights into practical deployment challenges and model robustness under varying clinical conditions.

5.3.2 Future Research Directions. Based on our findings, we identify several promising directions for future research [53, 55]. The

development of adaptive temporal resolution mechanisms could optimize the performance-efficiency trade-off [9] by dynamically adjusting time steps based on input complexity. Incorporating resource usage metrics into the training process could lead to more efficient models without sacrificing accuracy. Adding other medical signals to the system could help doctors make better diagnoses without slowing the analysis.

5.4 Broader Impact

Our approach's demonstrated performance and efficiency have implications beyond ECG classification [19]. The principles of our Siamese-SNN architecture could be adapted to other time-series classification tasks in healthcare [10], offering a promising framework for various biomedical signal processing applications. The system's low computing needs mean more healthcare facilities can access advanced heart monitoring technology, even with limited resources.

The results confirm that our method achieves better ECG classification accuracy without requiring excessive computing power. We found clear relationships between timing precision and resource needs, helping guide system setup in different clinical settings. At the same time, the model's scalability and robust performance characteristics support its potential for widespread adoption in healthcare settings.

6 CONCLUSION

This paper presents a Siamese Neural Network architecture for ECG classification that effectively balances performance and computational efficiency [19]. Our experimental results demonstrate that the baseline configuration achieves 99.67% test accuracy with only 8-time steps while increasing temporal resolution to 16 steps yields a marginal improvement to 99.70% at the cost of 10% longer training time [37]. The model exhibits rapid convergence characteristics, reaching 98% accuracy within the first 5 epochs, and maintains stable training dynamics throughout the process.

The architecture's practical advantages are evident in its resource efficiency and scalability [21], making it suitable for deployment in various clinical settings [22, 56]. Our detailed resource monitoring confirms that the model maintains modest memory requirements while demonstrating linear scaling properties with batch size increases. Combined with the model's robust performance across different configurations, these characteristics suggest its viability for real-world clinical applications [8].

While current limitations include the need for validation on more diverse datasets [13] and potential for further resource optimization [9], the architecture shows promise for broader applications in biomedical signal processing. Future work focusing on adaptive temporal resolution and resource-aware training could enhance the model's utility in resource-constrained healthcare environments [55].

The proposed Siamese-SNN architecture demonstrates effective ECG classification performance [53], balancing computational efficiency with classification accuracy. The analysis of performance-resource trade-offs provides insights for future ECG classification system development [19].

REFERENCES

[1] U Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Hojjat Adeli. 2017. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Computers in biology and medicine* 100 (2017), 270–278.

[2] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. 2018. Long short-term memory and learning-to-learn in networks of spiking neurons. arXiv:1803.09574 [cs.NE] https://arxiv.org/abs/1803.09574

[3] Emelia J. Benjamin, Paul Muntner, Alvaro Alonso, Marcio S. Bittencourt, Clifton W. Callaway, April P. Carson, Alanna M. Chamberlain, Alexander R. Chang, Susan Cheng, Sandeep R. Das, Francesca N. Delling, Luc Djousse, Mitchell S.V. Elkind, Jane F. Ferguson, Myriam Fornage, Lori Chaffin Jordan, Sadiya S. Khan, Brett M. Kissela, Kristen L. Knutson, Tak W. Kwan, Daniel T. Lackland, Tené T. Lewis, Judith H. Lichtman, Chris T. Longenecker, Matthew Shane Loop, Pamela L. Lutsey, Seth S. Martin, Kunihiro Matsushita, Andrew E. Moran, Michael E. Mussolino, Martin O'Flaherty, Ambarish Pandey, Amanda M. Perak, Wayne D. Rosamond, Gregory A. Roth, Uchechukwu K.A. Sampson, Gary M. Satou, Emily B. Schroeder, Svati H. Shah, Nicole L. Spartano, Andrew Stokes, David L. Tirschwell, Connie W. Tsao, Mintu P. Turakhia, Lisa B. VanWagner, John T. Wilkins, Sally S. Wong, Salim S. Virani, On behalf of the American Heart Association Council on Epidemiology, Prevention Statistics Committee, and Stroke Statistics Subcommittee. 2019. Heart Disease and Stroke Statistics—2019 Update: A Report From the American Heart Association. *Circulation* 139, 10 (2019), e56–e528. https://doi.org/10.1161/CIR.0000000000000659 arXiv:https://www.ahajournals.org/doi/pdf/10.1161/CIR.0000000000000659

[4] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1994. Signature verification using a "Siamese" time delay neural network. *Advances in Neural Information Processing Systems* 6 (1994), 737–744.

[5] Tsung-Ming Chen et al. 2020. Deep learning for cardiac arrhythmia diagnosis. *IEEE Reviews in Biomedical Engineering* (2020).

[6] Davide Chicco. 2021. Siamese Neural Networks: An Overview. In *Methods in Molecular Biology*. Vol. 2190. Humana, 73–94. https://doi.org/10.1007/978-1-0716-0826-5_3

[7] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2005), 539–546.

[8] Gari D Clifford, Chengyu Liu, Benjamin Moody, Li-wei H Lehman, Ikaro Silva, Qiao Li, AE Johnson, and Roger G Mark. 2017. AF classification from a short single lead ECG recording: The PhysioNet/Computing in Cardiology Challenge 2017. *Computing in Cardiology* 44 (2017), 1–4.

[9] Lei Deng, Yujie Wu, Xing Hu, Ling Liang, Yufei Ding, Guoqi Li, Guangshe Zhao, Peng Li, and Yuan Xie. 2020. Rethinking the performance comparison between SNNS and ANNS. *Neural Networks* 121 (2020), 294–307. https://doi.org/10.1016/j.neunet.2019.09.005

[10] Weihua Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. 2020. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. *arXiv preprint arXiv:2007.05785* (2020).

[11] A. Fois and B. Girau. 2023. Enhanced representation learning with temporal coding in sparsely spiking neural networks. *Frontiers in Computational Neuroscience* (2023).

[12] Wulfram Gerstner and Werner M Kistler. 2002. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge University Press.

[13] Ary L Goldberger et al. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation* 101, 23 (2000), e215–e220.

[14] M. Grant, I. Abdelgawad, C. Lewis, and B. Zordoky. 2020. Lack of sexual dimorphism in a mouse model of isoproterenol-induced cardiac dysfunction. *PLoS One* 15, 7 (2020), e0232507.

[15] R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2. 1735–1742. https://doi.org/10.1109/CVPR.2006.100

[16] Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. 2019. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature Medicine* 25, 1 (2019), 65–69. https://doi.org/10.1038/s41591-018-0268-3

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV] https://arxiv.org/abs/1512.03385

[18] Shenda Hong, Yanbo Xu, Alind Khare, Satyananda Priyadarshini, Kritika Joshi, and Anurag Gogna. 2020. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. *IEEE Journal of Biomedical and Health Informatics* 24, 11 (2020), 3229–3239.

[19] Shenda Hong, Yuxi Zhou, Junyuan Shang, Cao Xiao, and Jimeng Sun. 2020. Opportunities and challenges of deep learning methods for electrocardiogram

data: A systematic review. *Computers in Biology and Medicine* 122 (2020), 103801. https://doi.org/10.1016/j.compbiomed.2020.103801

[20] P. Y. Hsiao and C. C. Lo. 2013. A Plastic Cortico-Striatal Circuit Model of Adaptation in Perceptual Decision. *Frontiers in Computational Neuroscience* (2013).

[21] Eunshim Jeon, Kyeongyeon Oh, Soonil Kwon, Hyeseon Son, Yeji Yun, Eun-Seok Jung, and Min-Seok Kim. 2020. A Lightweight Deep Learning Model for Fast Electrocardiographic Beats Classification With a Wearable Cardiac Monitor: Development and Validation Study. *JMIR Medical Informatics* 8, 3 (2020), e17037. https://doi.org/10.2196/17037

[22] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. 2015. Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE Transactions on Biomedical Engineering* 63, 3 (2015), 664–675.

[23] Serkan Kiranyaz, Turker Ince, and Moncef Gabbouj. 2016. Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks. *IEEE Transactions on Biomedical Engineering* 63, 3 (2016), 664–675. https://doi.org/10.1109/TBME.2015.2468589

[24] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. *ICML deep learning workshop* 2 (2015).

[25] Gregory R. Koch. 2015. Siamese Neural Networks for One-Shot Image Recognition. https://api.semanticscholar.org/CorpusID:13874643

[26] Pablo Laguna, Raimon Jané, and Pere Caminal. 1994. Automatic detection of wave boundaries in multilead ECG signals: Validation with the CSE database. *Computers and Biomedical Research* 27, 1 (1994), 45–60.

[27] Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. 2016. Training Deep Spiking Neural Networks using Backpropagation. arXiv:1608.08782 [cs.NE] https://arxiv.org/abs/1608.08782

[28] Yang Li and Yi Zeng. 2022. Efficient and Accurate Conversion of Spiking Neural Network with Burst Spikes. arXiv:2204.13271 [cs.NE] https://arxiv.org/abs/2204.13271

[29] Eduardo José da S Luz, William Robson Schwartz, Guillermo Cámara-Chávez, and David Menotti. 2016. ECG-based heartbeat classification for arrhythmia detection: A survey. *Computer Methods and Programs in Biomedicine* 127 (2016), 144–164.

[30] Wolfgang Maass. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural Networks* 10, 9 (1997), 1659–1671.

[31] Marco Melis, Ambra Demontis, Battista Biggio, Gavin Brown, Giorgio Fumera, and Fabio Roli. 2018. Towards evaluating the robustness of neural networks. *IEEE Symposium on Security and Privacy* (2018), 39–57.

[32] G.B. Moody and R.G. Mark. 2001. The impact of the MIT-BIH Arrhythmia Database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50. https://doi.org/10.1109/51.932724

[33] Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. 2019. Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine* 36, 6 (2019), 51–63. https://doi.org/10.1109/MSP.2019.2931595

[34] Shu Lih Oh, Eddie Y.K. Ng, Ru San Tan, and U. Rajendra Acharya. 2018. Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats. *Computers in Biology and Medicine* 102 (2018), 278–287. https://doi.org/10.1016/j.compbiomed.2018.06.002

[35] Stanislaw Osowski, Linh Tran Hoai, and Tomasz Markiewicz. 2004. Support vector machine-based expert system for reliable heartbeat recognition. *IEEE transactions on biomedical engineering* 51, 4 (2004), 582–589.

[36] Jiapu Pan and Willis J Tompkins. 1985. A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering* 3 (1985), 230–236.

[37] Priyadarshini Panda, Aparna Aketi, and Kaushik Roy. 2019. Towards Scalable, Efficient and Accurate Deep Spiking Neural Networks with Backward Residual Connections, Stochastic Softmax and Hybridization. arXiv:1910.13931 [cs.CV] https://arxiv.org/abs/1910.13931

[38] Saman Parvaneh, Jonathan Rubin, Saeed Babaeizadeh, and Minnan Xu-Wilson. 2019. Cardiac arrhythmia detection using deep learning: A review. *Journal of Electrocardiology* 57S (2019), S70–S74. https://doi.org/10.1016/j.jelectrocard.2019.08.004

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, Vol. 32. 8026–8037.

[40] Michael Pfeiffer and Thomas Pfeil. 2018. Deep Learning With Spiking Neurons: Opportunities and Challenges. *Frontiers in Neuroscience* 12 (2018), 774. https://doi.org/10.3389/fnins.2018.00774

[41] Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. 2017. Cardiologist-level arrhythmia detection with convolutional neural networks. *Nature Medicine* 25, 1 (2017), 65–69.

[42] Nitin Rathi and Kaushik Roy. 2020. DIET-SNN: Direct Input Encoding With Leakage and Threshold Optimization in Deep Spiking Neural Networks. arXiv:2008.03658 [cs.NE] https://arxiv.org/abs/2008.03658

[43] Antonio H. Ribeiro, Manoel H. Ribeiro, Gabriela M. M. Paixão, Derick M. Oliveira, Paulo R. Gomes, Jéssica A. Canazart, Milton P. Ferreira, Carl R. Andersson, Peter W. Macfarlane, Mário Wagner Jr, et al. 2020. Automatic diagnosis of the 12-lead ECG using a deep neural network. *Nature Communications* 11, 1 (2020), 1760. https://doi.org/10.1038/s41467-020-15432-4

[44] Priyadarshini Roy et al. 2019. Towards scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization. *Frontiers in Neuroscience* (2019).

[45] Sebastian Schmidt, Andreas Bauer, and Magnus Petersson. 2016. Introducing second-order attention to LSTM networks for ECG classification. *Computing in Cardiology* (2016), 1–4.

[46] Hooman Sedghamiz. 2014. Complete Pan-Tompkins implementation ECG QRS detector. *Matlab Central* (2014).

[47] Xiaoyu Song, Feng Liu, Jianqing Wang, et al. 2021. ECG arrhythmia classification using a convolutional transformer network. *Physiological Measurement* 42, 6 (2021), 065002.

[48] Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. 2019. Deep learning in spiking neural networks. *Neural Networks* 111 (March 2019), 47–63. https://doi.org/10.1016/j.neunet.2018.12.002

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need. arXiv:1706.03762 [cs.CL] https://arxiv.org/abs/1706.03762

[50] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. 2019. Few-shot learning: A survey. *Comput. Surveys* 53, 3 (2019), 1–34.

[51] Jos L Willems, C Abreu-Lima, P Arnaud, JH van Bemmel, C Brohet, R Degani, B Denis, J Gehring, I Graham, G van Herpen, et al. 1991. The diagnostic performance of computer programs for the interpretation of electrocardiograms. *New England Journal of Medicine* 325, 25 (1991), 1767–1773. https://doi.org/10.1056/NEJM199112193252503

[52] World Health Organization. 2021. Cardiovascular diseases (CVDs). *WHO Fact Sheets* (2021). https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[53] Desheng Wu et al. 2021. A parallel neural network approach to interest rate forecasting. *Neural Computing and Applications* 33 (2021), 3359–3371.

[54] Can Ye, B. V. K. Vijaya Kumar, and Miguel Tavares Coimbra. 2012. Heartbeat Classification Using Morphological and Dynamic Features of ECG Signals. *IEEE Transactions on Biomedical Engineering* 59, 10 (2012), 2930–2941. https://doi.org/10.1109/TBME.2012.2213253

[55] Friedemann Zenke and Tim P. Vogels. 2020. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *bioRxiv* (2020). https://doi.org/10.1101/2020.06.29.176925 arXiv:https://www.biorxiv.org/content/early/2020/06/29/2020.06.29.176925.full.pdf

[56] Özal Yıldırım, Paweł Pławiak, Ru-San Tan, and U. Rajendra Acharya. 2018. Arrhythmia detection using deep convolutional neural network with long duration ECG signals. *Computers in Biology and Medicine* 102 (2018), 411–420. https://doi.org/10.1016/j.compbiomed.2018.09.009