ADVANCING ARRHYTHMIA DETECTION IN MICRO ECG THROUGH CONVOLUTIONAL NEURAL NETWORKS

A project report submitted in partial fulfillment of the requirements for B.Tech. Project

Integrated Post Graduate (M.Tech)

by

Shubhang Gupta (2020IMT-098)

Under the supervision of

Dr. Sandesh Jain

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR - 474015

2023

CANDIDATE'S DECLARATION

I, **Shubhang Gupta (2020IMT-098)**, a student pursuing an Integrated Post Graduate in Masters of Technology (Information Technology), hereby affirm that the Project Report titled –**"Advancing Arrhythmia Detection in micro ECG Through Convolutional Neural Networks"**, submitted to the institution to fulfill the requirements for the Integrated Post Graduate in Masters of Technology - Information Technology degree, represents the genuine outcome of my independent research conducted between *May 2023* and *August 2023* under the guidance of **Dr. Sandesh Jain**. Proper citations have been provided for all referenced text, figures, and tables.

Date: Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: Signature of the Supervisor

ABSTRACT

The Electrocardiogram (ECG) serves as a crucial clinical signal extensively employed for the detection of cardiovascular ailments. Nevertheless, the manual assessment of ECG signals can become laborious, particularly when dealing with a substantial number of cardiac patients due to subtle physiological variations. To alleviate this burden, the automated classification of ECG signals can be a valuable tool for healthcare professionals in making precise diagnoses. Our study introduces a classification model founded on data-driven non-linear features, extracted through a 1D-CNN architecture. This model efficiently categorizes ECG signals into five distinct classes, encompassing Non-ectopic beats (Normal Beat), Supraventricular ectopic beats, Ventricular ectopic beats, Fusion Beats, and Unknown Beats. Following a rigorous 5-fold cross-validation, the proposed algorithm demonstrates remarkable accuracy, achieving approximately 97.36% accuracy and an f1 score of around 99.83%. This presents a straightforward and swift model that can be readily implemented on e-healthcare devices, facilitating remote heart diagnosis.

Cardiovascular disease (CVD) continues to be a leading cause of mortality, underscoring the critical need for effective arrhythmia detection through non-invasive ECG analysis. Automated computer-assisted techniques, such as convolutional neural networks (CNNs), play a pivotal role in mitigating the challenges associated with manual ECG analysis, particularly when dealing with data from Holter monitors. Our modified 12-layer deep one-dimensional CNN addresses the limitations of traditional CNN methodologies, ensuring enhanced efficiency while preserving classification accuracy. This research represents a significant contribution to the field of cardiac rhythm analysis, showcasing the potential of streamlined deep learning models in achieving efficient arrhythmia detection within micro ECG data.

Keywords - ECG, Micro-classes, Complexity, MIT-BIH, CNN, Denoise, 12 layer deep 1D CNN, Wavelet, ReLU, Heart disease, Deep learning, Automatic diagnosis

ACKNOWLEDGEMENT

Acknowledging the contributions of those who played pivotal roles in the successful completion of any endeavor is not just a formality; it is an essential gesture of recognition for their invaluable support, without which the project's fruition would have remained a mere theoretical concept.

First and foremost, I extend my heartfelt gratitude to **Prof. Aditya Trivedi**, the Head of the Department of Information Technology at ABV-IIITM Gwalior, along with all the esteemed faculty members of our department. Their unwavering guidance, motivational words, and genuine support throughout this project have been instrumental in its success. I am profoundly thankful for their academic mentorship and their unwavering belief in my abilities, which has bolstered my confidence significantly. It is through their support and encouragement that I was able to navigate this journey successfully.

I would like to express my appreciation to my dedicated supervisor, **Dr. Sandesh Jain**, Assistant Professor in the Department of Information Technology at ABV-IIITM. His generosity in granting me the freedom to explore innovative ideas, coupled with his guidance and confidence in my project, has been invaluable. His patience and openness to dialogue have made me feel like a seasoned professional rather than a novice. His unwavering interest and support have played a pivotal role in bringing my work to its current stage of accomplishment.

Lastly, I extend my gratitude to my institution and my colleagues whose continuous encouragement served as a constant source of inspiration, reinvigorating my determination and focus, thus facilitating the successful execution of this project.

Shubhang Gupta

Contents

List of Tables

List of Figures

LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE

1D	One-Dimensional
AI	Artificial Intelligence
AP	Atrial Premature
CNN	Convolution Neural Network
CPU	Central Processing Unit
CSV	Comma-Separated Values
DL	Deep Learning
DNN	Deep Neural Network
ECG	Electrocardiogram
EEG	Electroencephalogram
EMG	Electromyogram
EHR	Electronic Health Record
EMR	Electronic Medical Record
FDA	Food and Drug Administration
HIPAA	Health Insurance Portability and Accountability Act
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
LBBB	Left Bundle Branch Block
ML	Machine Learning
MIT-BIH	Massachusetts Institute of Technology-Beth Israel Hospital
NOR	Normal
PVC	Premature Ventricular Contraction
RBBB	Right Bundle Branch Block
ReLU	Rectified Linear Unit

Chapter 1

INTRODUCTION

1.1 Overview

The global population continues to grow steadily, leading to challenges in providing adequate medical services. The sheer numbers make it infeasible for every patient to have constant access to a doctor's care. In this context, time efficiency and rapid response are crucial for healthcare professionals. This project aims to address these challenges by integrating with existing tools, particularly at the edge of medical technology, minimizing the reliance on additional devices such as smartwatches. Additionally, this solution has potential applications in medical training.

Cardiovascular disease is a prevalent health concern, particularly affecting the well-being of middle-aged and elderly individuals. It is characterized by its high incidence, substantial rates of disability, and significant mortality rates. Given the global trend toward an aging population, the escalation of cardiovascular diseases has become a critical public health issue [1]. Electrocardiogram (ECG) analysis serves as a crucial tool for evaluating heart health [2]. Consequently, the understanding and precise classification of ECG signals play a pivotal role in comprehending heart-related conditions. This goes beyond early diagnosis and appropriate treatment; it also encompasses early preventive measures [3] [4].

1.2 The Electrocardiogram

In the past, medical practitioners primarily relied on invasive procedures to diagnose and treat internal organ disorders, often subjecting patients to continuous pain and discomfort. This approach posed significant challenges for both doctors and patients alike. Fortunately, modern technology has revolutionized medical practices, ushering in non-invasive techniques such as blood tests and lab assessments. These methods allow doctors to gather valuable data from various organs within the human body, contributing to a broader spectrum of diagnostic information. Examples of these signals include EMG (Electromyogram, measuring electrical activity in muscles), EEG (Electroencephalogram, assessing electrical brain activity), and ECG (Electrocardiogram, monitoring the heart's electrical activity). This project, specifically, focuses on the analysis of ECG signals, providing further insights into this particular signal.

ECG signal analysis involves studying the heart's electrical processes, including depolarization (positive electrical activation of heart tissues) and repolarization (discharge of electrical charge from these tissues). An array of strategically positioned electrodes on the body's surface is used to capture this cardiac electrical activity, generating data known as Electrocardiograms, abbreviated as ECG or EKG. Recently, there has been a notable shift towards employing microcomputers for ECG data processing, showing significant promise in enhancing healthcare practices.

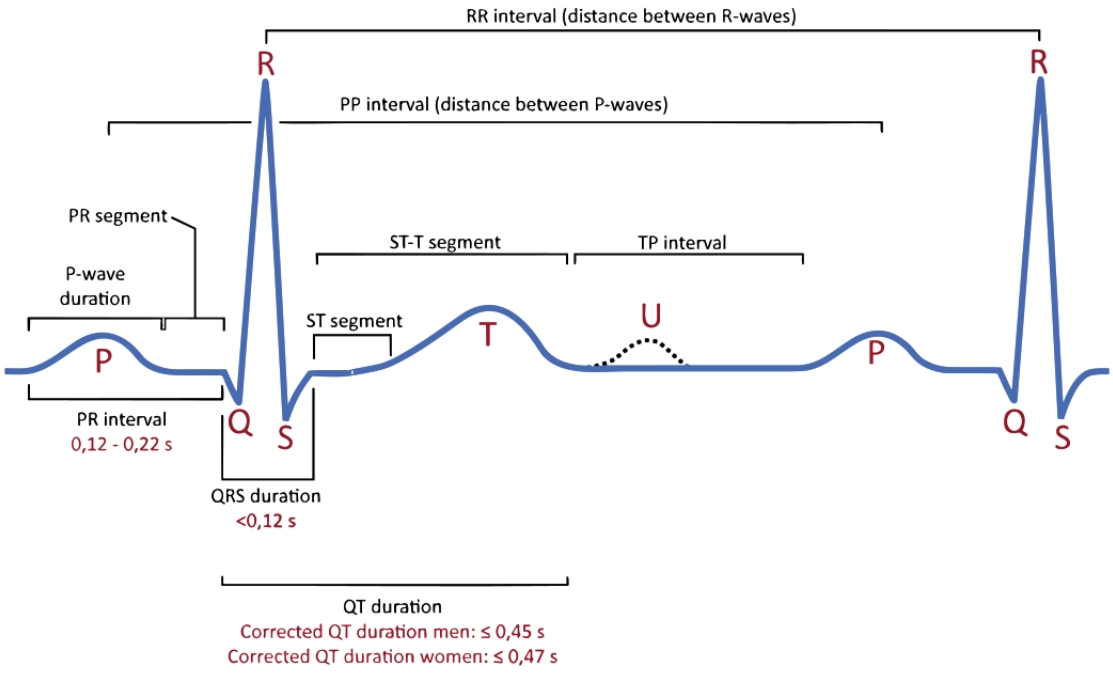

Figure 1.1: Classical ECG signal characteristics

Figure 1.1 provides an analysis of the temporal dynamics of the heart's rhythmic cycles, captured in the form of typical ECG waveforms. The figure exudes precision, notably depicting periodic spikes that coincide with ventricular contractions, scientifically termed "depolarization." During depolarization, muscle fibers shed their resting potential. Signals that would typically undergo a minor negative deviation instead exhibit a prominent positive incline, followed by a subsequent negative deviation. This ensemble is recognized as the "QRS complex," representing the ventricular contraction. Subsequent to the QRS complex, a discernible oscillation emerges, often referred to as the ST segment or ST wave. At this juncture, the potential difference becomes indeterminate, marked by an isoelectric line, also known as the zero line. The T-wave ensues, signifying the depolarization phase. Concluding the sequence, the P-wave signifies atrial excitation, while the U-wave, although typically diminutive, is indicative of the repolarization of papillary muscles. Figure 1.2, on the other hand, presents an ECG signal, albeit purely hypothetical in nature. It's essential to note that the genuine ECG signal exhibits greater complexity compared to this smoothed representation, emphasizing the intricacies involved in cardiac monitoring and analysis.

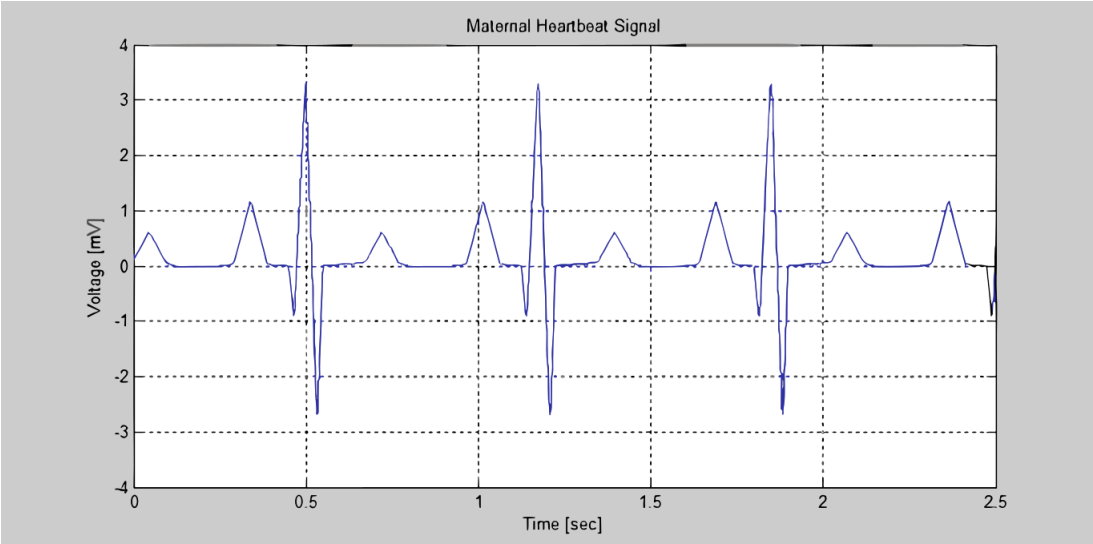

Figure 1.2: Shape of electrocardiogram

An Electrocardiogram (ECG), as exemplified in Figure 1.3, is a diagnostic tool that records electrical impulses originating from the heart muscle's surface, capturing data from various spatial perspectives. It offers a comprehensive assessment of the heart's electrical activity. This intricate process involves the generation of action potentials within the heart muscle during contractions, which facilitate the crucial task of pumping blood throughout the entire body. It is imperative to note that this mechanical contraction within the heart muscle inherently produces electrical activity as a consequential aspect of its function.

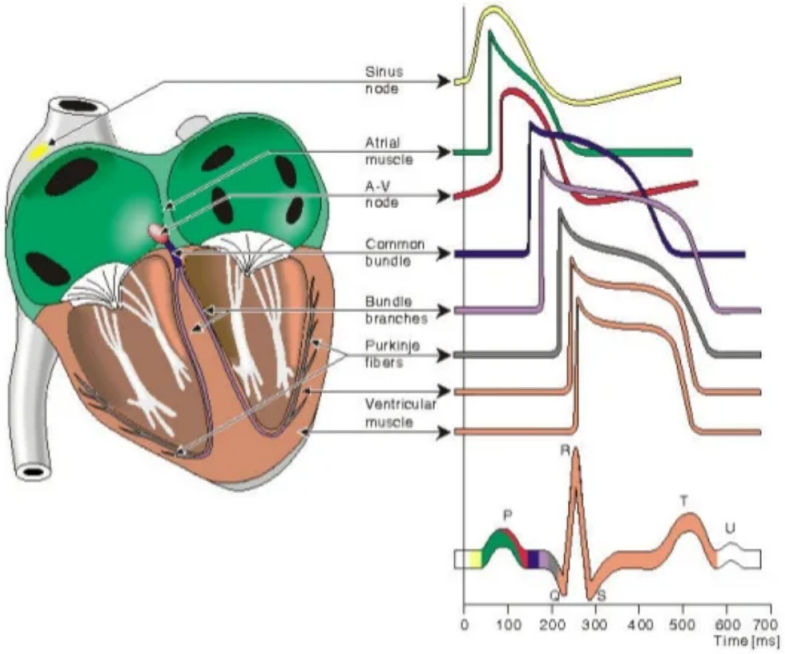

Figure 1.3: Heart's anatomy with different waveforms

1.3 ECG Waveform Description

The ECG waveform is generated as a representation of the condensed potential vectors of the heart. It comprises several peaks and distinctive patterns, which play a crucial role in waveform analysis and diagnosis. These distinctive features include:

1. The P-wave: This waveform signifies the propagation of depolarization from the SA (Sinoatrial) node to the atria, with a typical duration of 80–100 milliseconds.

2. P-R interval: This interval represents the time it takes for an electrical impulse to travel from the SA node to the AV (Atrioventricular) node before gaining access to the ventricles, with a duration ranging from 120 to 200 milliseconds.

3. P-R segment: Corresponding to the period between the conclusion of atrial depolarization and the onset of ventricular depolarization, this segment typically lasts about 100 milliseconds.

4. QRS complex: This complex reflects the depolarization of the ventricles. Normally, its duration falls within the range of 0.06 to 0.1 seconds.

5. Q-wave: Signifying the usual left-to-right depolarization of the interventricular septum.

6. R-wave: Represents the early depolarization of the ventricles.

7. S-wave: Represents the late depolarization of the ventricles.

8. S-T segment: This segment follows the QRS complex and indicates the complete depolarization of the ventricles.

9. Q-T interval: An estimation of the typical duration of ventricular action, encompassing both repolarization and depolarization. Its length can vary with heart rate and typically ranges from 0.2 to 0.4 seconds.

10. T-wave: Indicates that the duration of ventricular repolarization is greater than that of depolarization.

1.4 Motivation

The landscape of cardiovascular ailments presents a diverse array of manifestations, and the protracted process of manual diagnosis carries an inherent risk of yielding spurious positive results. In the contemporary medical landscape, a pressing challenge looms: the need to achieve precise and expeditious analyses of specific cardiac conditions. Compounding the complexity of this endeavor is the erratic, low-frequency, and delicate nature of electrocardiogram (ECG) signals, which renders diagnostic outcomes vulnerable to instability. In pursuit of elevating the effectiveness and precision of ECG interpretation, there exists an unequivocal imperative for advanced, automated identification, and classification of ECG signals [5].

This study undertakes a comprehensive exploration into the realm of micro-classification, focusing on discrete cardiac contractions: the Normal (NOR) rhythm, Left Bundle Branch Block (LBBB), Right Bundle Branch Block (RBBB), Atrial Premature (AP) contractions, and Premature Ventricular Contraction (PVC) anomalies. The impetus behind this meticulous approach stems from the relative scarcity of efforts dedicated to dissecting these subtle nuances embedded within ECG signals. Additionally, a conspicuous gap persists

in the field concerning endeavors aimed at alleviating the computational burdens associated with existing models, thereby rendering them amenable for deployment on resource-constrained edge computing devices.

Hence, the driving force behind this undertaking lies in forging a paradigm that not only elevates diagnostic precision but also effectively mitigates computational complexity, achieving a harmonious equilibrium between precision and efficiency. To accomplish this, a dual-pronged strategy was adopted. The initial steps involved the application of a wavelet-driven self-adaptive threshold methodology to meticulously eliminate detrimental noise from the raw ECG signal, thereby establishing a pristine foundation for rigorous analysis. Subsequent to this preparatory phase, we embarked on creating an innovative iteration of a 12-layer deep one-dimensional convolutional neural network (CNN) architecture. Remarkably, this architectural innovation is characterized by its astute reduction in trainable parameters, a significant departure from the configuration proposed in a preceding study [6].

The crux of this endeavor lies in the meticulous micro-classification of ECG signals meticulously sourced from the MIT-BIH Arrhythmia database. By skillfully navigating the convergence of sophisticated signal processing techniques and inventive neural network design, our aspiration is to herald a new era of precision and discernment in the diagnosis of arrhythmias. This dual-edged pursuit of precision enhancement and computational frugality paves the way for a domain where the identification of cardiac irregularities achieves unprecedented accuracy while concurrently shedding the burden of onerous computational demands.

Chapter 2

LITERATURE REVIEW

2.1 Background

In the wake of AI-driven technological advancements, a plethora of machine learning techniques are being harnessed to tackle the challenges posed by the vast troves of ECG signal data and the labor-intensive nature of manual detection. In concert with remarkable progress in diverse domains such as image processing, audio recognition, and numerous others [7] [8] [9] [10], the realm of machine learning and deep learning networks has witnessed extensive adoption for the aided detection of cardiac conditions based on ECG data [11] [12] [13].

Deep learning networks, celebrated for their capacity to discern complex data patterns, autonomously extract features, and eliminate the necessity for intricate signal preprocessing, have assumed a central role in this field. Additionally, their exceptional ability for nonlinear fitting provides them with a distinct advantage in handling single-lead, multi-class, and imbalanced ECG datasets [12]. Among the various neural architectures, the Convolutional Neural Network (CNN), a type of feedforward neural network that has been extensively researched and applied in the realm of deep learning, emerges as a powerful tool for the efficient classification of arrhythmic ECG signals [11] [12] [13].

The march of technological progress, driven by the capabilities of AI, has ushered in a transformative era in the field of ECG signal analysis. It is within this landscape that machine learning and deep learning networks emerge as powerful allies, illuminating pathways toward more efficient and precise detection of cardiac maladies.

2.2 Key Related Research

- **Xu, Xuexiang and Liu, Hongxing** in their paper titled ”ECG Heartbeat Classification Using Convolutional Neural Networks” shed light on the challenges associated with manually categorizing heartbeats. Manual classification not only presents formidable analytical difficulties but also consumes substantial time. Moreover, manual classification often results in erroneous arrhythmia detections. Thus, there arises a compelling need for automated computer-aided methods to scrutinize and categorize each heartbeat. Convolutional Neural Networks (CNNs) emerge as the solution, automating ECG signal classification without the need for laborious manual feature extraction [14].

- **Tao Wang, Changhua Lu, Yining Sun, Mei Yang, Chun Liu, and Chunsheng Ou** propose an innovative approach in their paper, ”Automatic ECG Classification Using Continuous Wavelet Transform and Convolutional Neural Network.” Their methodology harnesses the power of Continuous Wavelet Transform (CWT) in conjunction with Convolutional Neural Network (CNN). Here, CNN extracts features from the 2D-scalogram formed by CWT’s decomposition of ECG signals. Notably, four RR interval characteristics, coupled with CNN features, are fed into the dense layer for ECG classification. This approach emphasizes the significance of the surrounding R peak interval (RR interval) in arrhythmia detection [15].

- **Min Lin, Qiang Chen, and Shuicheng Yan** introduce the concept of ”Network In Network” in their paper. They advocate for the utilization of Global Average Pooling instead of traditional fully connected layers within standard convolutional neural networks. This alteration encourages correlation among feature maps and categories, facilitated by a more robust local modeling through the micro network. Global Average Pooling is not only relevant but also inherently acts as a structural regularizer, guarding against overfitting in contrast to dense layers, which are susceptible to overfitting and often rely on dropout regularization [16] [17].

- **Mengze Wu, Yongdi Lu, Wenli Yang, and Shen Yuong Wong** address the shortcomings of prior research, which involve the manual identification of features, the utilization of intricate models, and extended training durations. Their study is centered on the micro-classification of Electrocardiogram (ECG) signals, specifically

targeting five subcategories of heartbeat types found within the MIT-BIH Arrhythmia database. They introduce a dependable and efficient 12-layer one-dimensional convolutional neural network (1D CNN) for the classification of these five distinct heartbeat characteristics. This model benefits from the wavelet self-adaptive threshold denoising technique [6].

- **Sabrina Göllner's** article explores strategies for reducing trainable parameters in CNNs while maintaining accuracy levels above 99%. These approaches include the use of MaxPooling to downsample input for subsequent convolutional layers, adopting global average pooling to reduce trainable parameters, and applying pruning techniques to trim layers, filters, and kernel sizes for parameter reduction.

- **Masko, David, and Hensman's** study, "The Impact of Imbalanced Training Data for Convolutional Neural Networks," delves into the consequences of imbalanced training data on CNN performance in image classification. Their findings underscore the viability of oversampling as a countermeasure to mitigate the effects of data imbalances during training.

- **Rakshith Vasudev** elucidates the mechanics of CNN learning in his article, "Understanding and Calculating the Number of Parameters in Convolution Neural Networks (CNNs)." He provides an elaborate account of how various layers, including the input layer, CNN layer, Pool layer, Fully Connected layer, and softmax layer, contribute to trainable parameters in CNNs.

- Within the insightful treatise penned by **Kizito Nyuytiymbiy**, the focus is firmly fixed on the realms of "Parameters and Hyperparameters in the Domain of Machine Learning and Deep Learning." This illuminating discourse elucidates the pivotal divergence between parameters and hyperparameters. In the intricate domains of deep learning and machine learning, a hyperparameter is, in essence, any facet whose values or configurations are thoughtfully selected prior to the initiation of the training process. These meticulously chosen values and settings remain steadfast, unwavering throughout the entirety of the training regimen, distinct from the adaptable parameters that dynamically evolve and adapt during the learning process.

2.3 Research Gaps

- To increase the accuracy of model less, attention was given to the complexity of model and models are highly complex and needs more computation and training time.
- Less attention was given towards classification of micro classes present in MIT-BIH arrhytmia dataset.
- Less use of global average pooling, instead fully connected layers are used in ECG classification which leads to overfitting.
- The previous model [6] was highly complex, hence not suitable for edge computing.

Chapter 3

Objectives and Deliverables

This chapter will discuss the objectives and deliverables. These are fundamental components that help define the purpose and scope of your work. And Easily explain what exactly are we doing in the project. Research Goals, Focus and Direction, Guidance, Evaluation, Communication, Tangible Outputs, Completion Milestones, Documentation, Presentation, Evaluation Criteria are some of the main points highlighted in this section.

3.1 Problem Formulation

In this Bachelor's Thesis Project (BTP), the core objective is to optimize the Micro ECG classifier, emphasizing both accuracy enhancement and efficiency. This entails a meticulous overhaul of the 12-layer 1D CNN architecture, incorporating advanced techniques like pooling, softmax, and dropout layers. Successful completion of this endeavor aims to yield a more precise classifier vital for accurate heartbeat classification and early arrhythmia detection.

The Micro ECG Classification project is rooted in deep learning, spotlighting a 12-layer deep 1D Convolutional Neural Network (CNN) within the context of edge computing. The primary research goals encompass the streamlining of the deep CNN architecture to suit resource-constrained edge environments. Subsequently, it aims to construct a resilient beat classification model grounded in deep learning principles. This model, a focal point of the research, holds the key to precise and efficient heartbeat classification, crucial for diagnosing cardiac arrhythmias. Further, the project includes data enhancement via the wavelet transform for raw ECG signal denoising, and it strives for micro-classification

of heartbeats into categories including Normal, LBBB, RBBB, AP, and PVC.

These objectives collectively underpin the development of an edge-deployable, deep learning-based ECG classification system, adept at efficiently processing raw ECG signals and categorizing them into micro-classes. This technology carries the potential to expedite real-time arrhythmia detection, offering timely medical insights and enhancing patient care within resource-constrained environments. Throughout the project, clear guidance, well-documented deliverables, and established evaluation criteria will ensure a structured investigation while facilitating effective communication of results and milestones.

3.2 Objectives

- The primary aim is to simplify the intricate 12-layer deep 1D Convolutional Neural Network (CNN) architecture, taking a significant stride towards optimizing it for edge computing environments.

- The central objective is the creation of a sophisticated beat classification model employing deep learning techniques, specifically harnessing the power of Convolutional Neural Networks (CNNs).

- Another pivotal goal is to enhance the clarity of raw Electrocardiogram (ECG) signals by applying the wavelet transform method for effective denoising.

- The research endeavors to achieve precise micro-classification of heartbeat types within the MIT-BIH Arrhythmia database, including Normal (NOR), Left Bundle Branch Block (LBBB), Right Bundle Branch Block (RBBB), Atrial Premature Beats (AP), and Premature Ventricular Beats (PVC). This micro-classification contributes significantly to the depth of analysis.

Chapter 4

System Architecture & Methodology

The methodology is a relevant structure for research. In this chapter, we have multiple sections that covers the Graphical abstract, Network architecture, proposed hypothesis, About dataset, pre-processing, Data segmentation, Data enhancement, Understanding parameters, hyperparameters and globalaveragepooling, mechanism, Implementation details.

4.1 Algorithm

- **Dataset Collection and Preparation:** To commence, acquiring the prestigious MIT-BIH Arrhythmia Database is essential. This database houses a comprehensive collection of electrocardiogram (ECG) recordings. Following this acquisition, meticulous preprocessing is imperative. This phase involves harmonizing ECG signals through resampling, a process designed to rectify any sampling rate discrepancies and establish a uniform rate, typically set at 360 Hz. Additionally, each data point must undergo meticulous labeling. This step is paramount as it imparts identities to the signals, associating them with their respective arrhythmia classes.

- **Data Splitting:** Having elegantly prepared the dataset, the subsequent task is to judiciously partition it into three cohorts: the training, validation, and test sets. Prudence is essential in this step, as it demands the execution of a partitioning strategy that ensures balance. Such balance guarantees an equitable distribution of arrhythmia classes within each set, thereby mitigating the risk of bias.

- **Model Architecture Design:** At the core of this undertaking lies the creation of a profound 12-layer deep Convolutional Neural Network (CNN), representing a testament to computational sophistication. The model's inception involves the instantiation of an input layer, enriched with critical parameters. These parameters encompass a kernel size of 9, a filter size of 16, strides set to 1, and the activation function of Rectified Linear Unit (ReLU). Additionally, the input shape is set at 360. Subsequent layers are meticulously fashioned, incorporating convolutions, pooling, dropout mechanisms, and dense strata. The architectural decisions are guided by the intricacies delineated in the aforementioned modifications. Notably, convolutional layers are adorned with the ReLU activation function, while the second Dense layer yields its position to a GlobalAveragePooling1D layer, thereby fostering feature extraction capabilities. The model culminates elegantly with a softmax output layer.

- **Model Compilation:** With architectural finesse in place, the model compilation phase commences. An astute selection of a loss function is of paramount importance, with categorical cross-entropy emerging as the preferred choice in this scenario. The optimization engine of choice is the distinguished Adam optimizer, fine-tuned with meticulously calibrated learning rates. Furthermore, a suite of evaluation metrics, including accuracy, serves as guiding beacons for assessing model performance.

- **Model Training:** The nucleus of this expedition, model training, is initiated by infusing data from the training dataset. Early stopping, a mechanism designed to mitigate overfitting, vigilantly monitors validation loss. Dropout, a valuable tool in the battle against overfitting, is judiciously applied during training to bolster model robustness.

- **Model Evaluation:** Upon the culmination of training, the model's mettle is tested using the validation dataset. An array of performance metrics, encompassing accuracy, precision, recall, and F1-score, rigorously scrutinizes its classification capabilities.

- **Hyperparameter Tuning (Optional):** If the quest for optimization demands it, a foray into hyperparameter exploration may be undertaken. This entails a meticu-

lous examination of kernel and filter sizes, with the ultimate aim of achieving peak model performance.

- **Model Testing:** The final arbiter of success resides in the testing phase. The model's prowess is tested when subjected to the hitherto unseen test dataset. This represents a crucible that elucidates its generalization capabilities.
- **Results Analysis:** With performance duly assessed, the ensuing task is result analysis. The model's proficiency in the classification of micro ECG arrhythmias is subject to scrutiny. The venerated tools of confusion matrices and ROC curves play a pivotal role in offering panoramic insights into its discriminatory abilities.

4.2 Modified 12 layer deep 1D CNN

The modified architecture consists of three parts, Figure 3.2:

- Raw ECG signal preprocessing, which includes signal denoising, data segmentation and then data enhancement.
- Then feeding the processed ECG signal to the proposed model.
- Finally classification output of five-micro heartbeats will be received at end.

Algorithm: 12 layer deep 1D CNN

1) **Input:** Load MIT-BIH Arrhythmia Database

2) **Output:** Estimated Model

3) **Initialize:** hyperparameters: $epochs$, $learning\_rate$, $dropout\_rate$, ε

4) **Initialize** model architecture: $model$

5) **Initialize** performance metrics: $best\_accuracy$, $best\_model$

6) **while** not converged:

7) Data Collection and Preprocessing

8) Collect ECG recordings X and labels Y from D

9) Resample X to 360Hz

10) Label X with arrhythmia classes in Y

11) Sample a batch of transitions from replay buffer R

12) **Update** $epochs$, $learning\_rate$, batch size

13) CNN Testing & Training

Compute the loss function:

$$Loss(X,Y,\theta) = \frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i(\theta))^2$$

Update model weights using gradient descent:

$$\theta \leftarrow \theta - learning\_rate \cdot \nabla Loss(X,Y,\theta)$$

14) Parameter Update and Optimization

Update dropout rate: $dropout\_rate \leftarrow new\_dropout\_rate$

Update other hyperparameters

15) **while** not converged

Convergence criterion: $|best\_accuracy - current\_accuracy| < \varepsilon$

16) Plot graphs & display result

17) Print Results and graphs

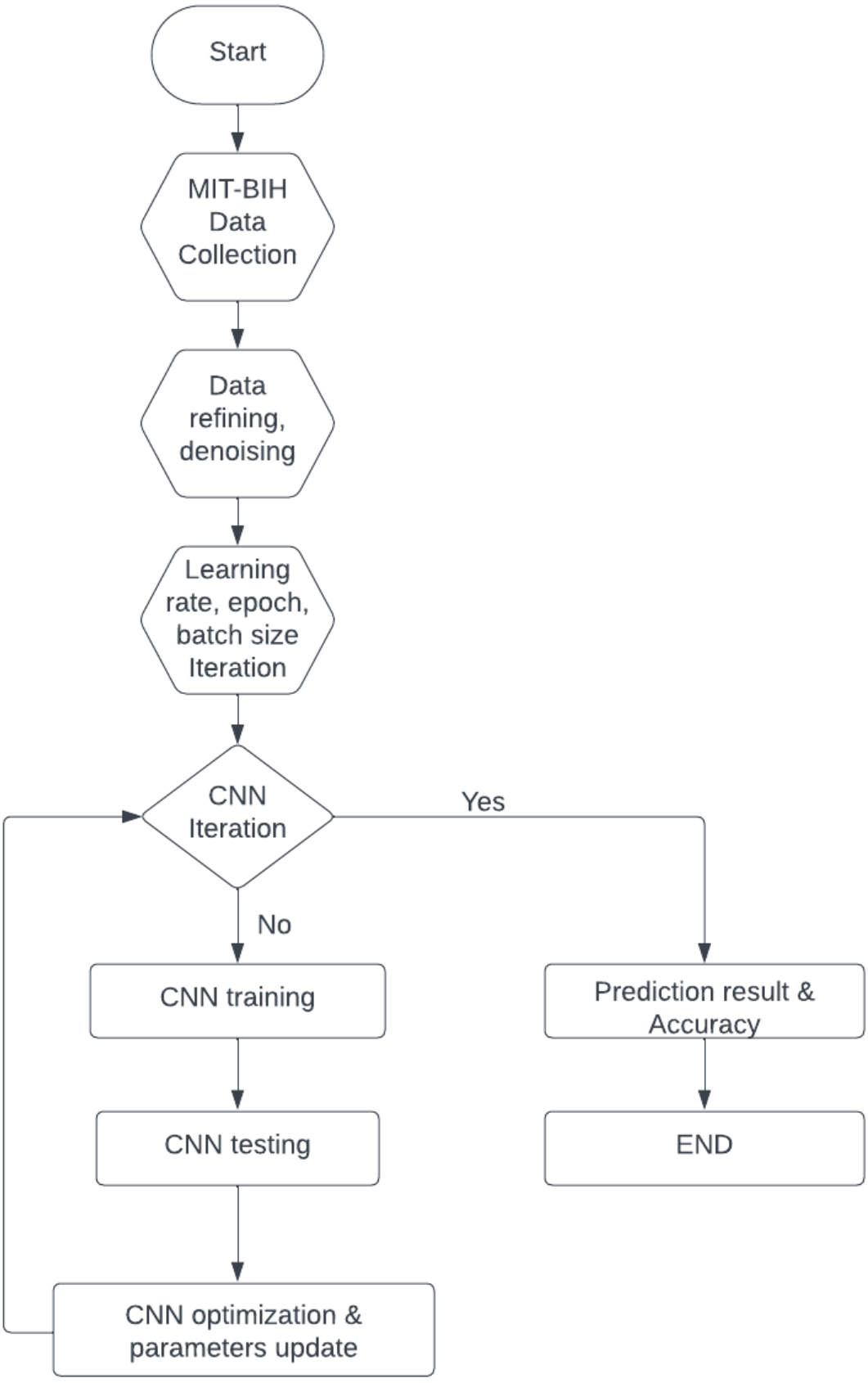

Figure 4.1: Basic workflow of the Algorithm

4.3 Graphical Abstract

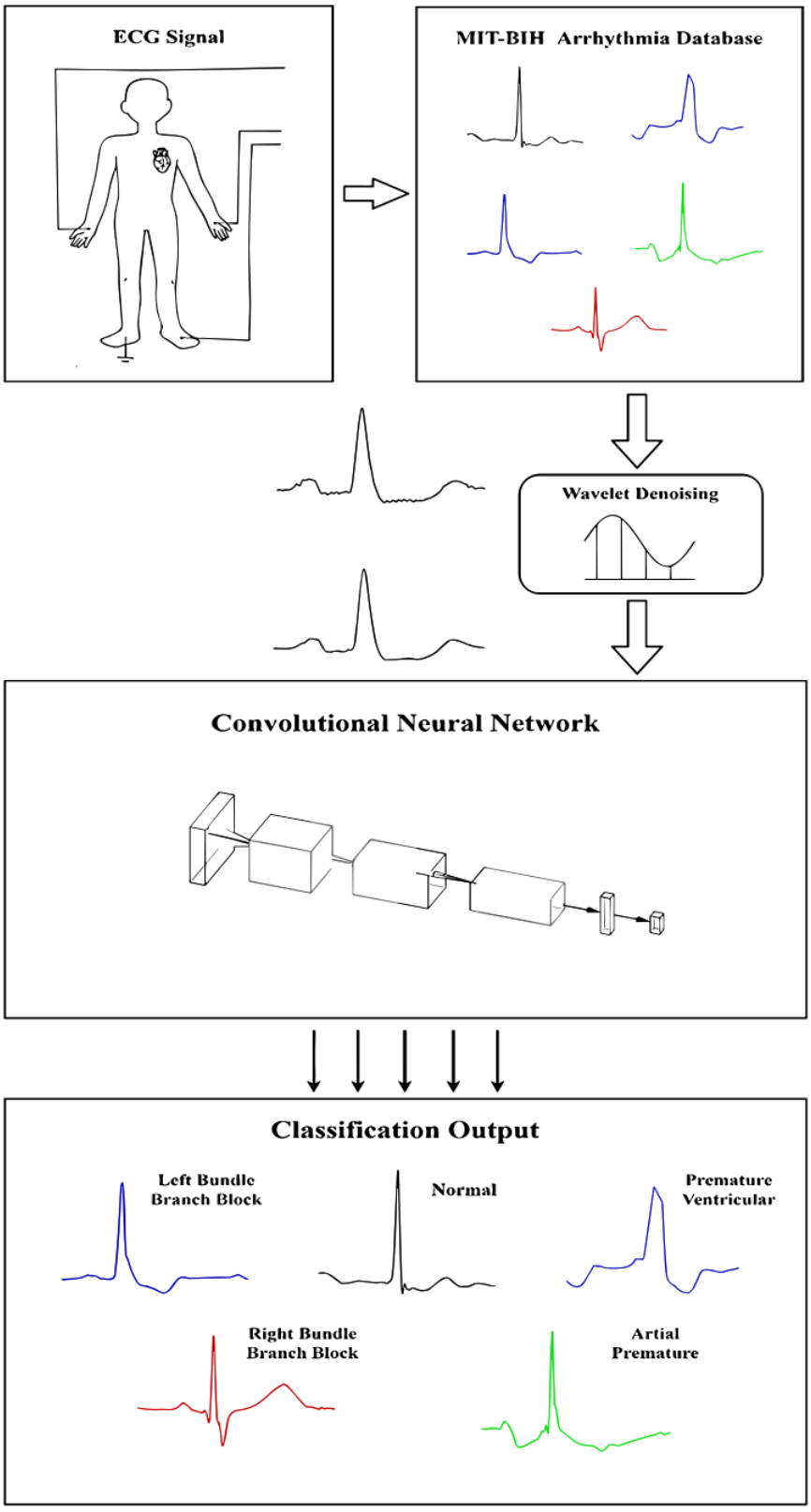

Figure 4.2: Graphical abstract of the methodology

4.4 Proposed Hypothesis

Pruning of hyperparameters (kernel and filter size) was utilised to reduce the number of trainable/learnable parameters, and Global Average Pooling 1D was used in place of dense or completely connected layers.

The act of pruning involves deleting weighty connections from a network. The over-parameterized network is de-parameterized in this process. It is done to quicken inference and reduce the size of model storage. Kernel size and filter size in the first four CNN layers are hyperparameters that are pruned in the model.

$$\mathbf{P} = (K * I + 1) * F$$

where P = Number Of Parameters In A Layer, K = Kernel size, I = Input channels, F = Number of filters and 1 is bias.

Global Average Pooling (GAP): In conventional classification processes, the final convolutional layer's feature maps are typically flattened and funneled through fully connected layers, followed by a softmax logistic regression layer. However, the employment of dense layers often invites overfitting concerns, necessitating the use of dropout regularization (0.5), which selectively deactivates half of the neuron activations within fully connected layers during training to mitigate overfitting risks.

In contrast, our approach incorporates global average pooling [18] in lieu of dense layers. This innovation entails the generation of a distinct feature map for each classification category within the final convolution layer. The essence of global average pooling lies in the direct averaging of each feature map, thereby producing a vector that seamlessly feeds into the softmax layer. This approach eliminates the need for intricate fully connected layers atop feature maps while naturally preserving their alignment with specific categories. Consequently, feature maps can be conceived as diverse confidence maps, offering a more intrinsic synergy with the convolutional structure. Notably, global average pooling also serves as an effective overfitting deterrent, given its parameterless nature. Furthermore, this technique exhibits resilience against spatial translations, as it amalgamates spatial data during processing.

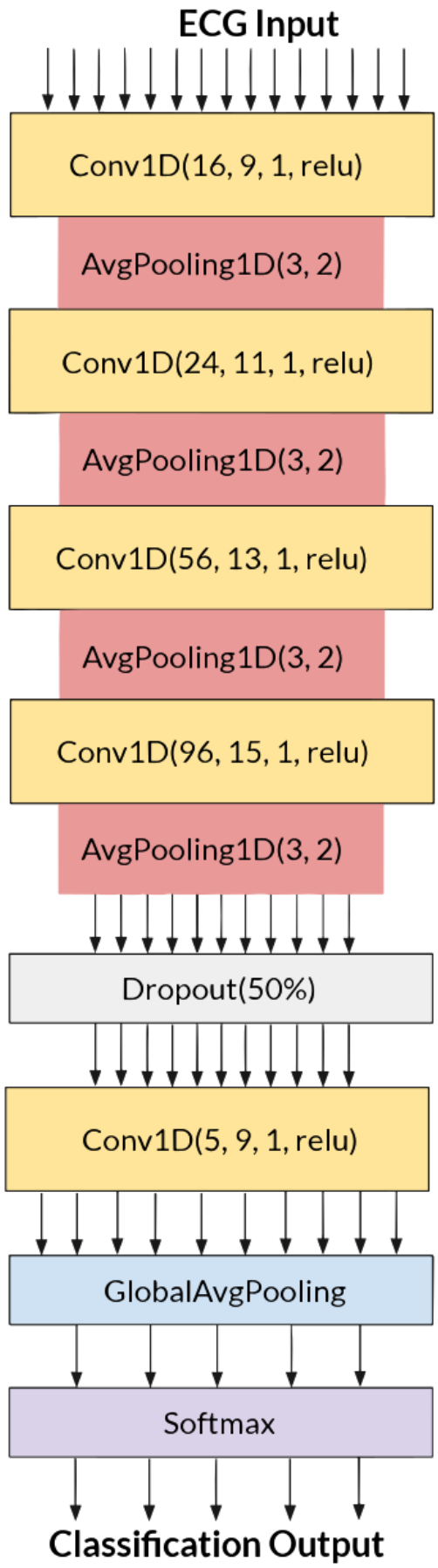

Figure 4.3: Modified CNN model

4.5 Dataset

The study relies on the MIT-BIH arrhythmia database, comprising 48 records of two-channel ECG signals spanning 30 minutes. These records were thoughtfully selected from extensive 24-hour recordings of 47 individuals, totaling an impressive 109,446 unique sample numbers. Among these participants, 25 were male, aged 32 to 89, and 22 were female, aged 23 to 89. Interestingly, records 201 and 202 are from the same male subject, introducing complexity. Each recording includes two leads: modified limb lead II and one of modified leads V1, V2, V4, or V5. ECG signals undergo rigorous band-pass filtering (0.1–100 Hz) and are digitized at 360 Hz. Notably, 23 recordings (numbered 100-124) represent routine clinical cases, while 25 encompass diverse ventricular, junctional, and supraventricular arrhythmias. The database includes meticulously verified annotations for timing and beat class information by independent experts.

1. Number of Samples: 100012
2. Number of Categories: 5
3. Sampling Frequency: 360 Hz
4. Major Categories: ['N': 0, 'S': 1, 'V': 2, 'F': 3, 'Q': 4]

Table 4.1: MIT-BIH Database based on Association for the Advancement of Medical Instrumentation

Category	**Classes**
N	• Normal beat (N) • Left and right bundle branch block beats (L,R) • Atrial escape beat (e) • Nodal (junctional) escape beat (j)
S	• Atrial premature beat (A) • Aberrated atrial premature beat (a) • Nodal (junctional) premature beat (J) • Supraventricular premature beat (S)
V	• Premature ventricular contraction (V) • Ventricular escape beat (E)
F	• Fusion of ventricular and normal beat (F)
Q	• Paced beat (/) • Fusion of paced and normal beat (f) • Unclassifiable beat (U)

From Table 4.1 the five micro classes that we will be using for this project are:

- Normal (NOR), symbol (.)
- Left bundle branch block (LBBB), symbol (L)
- Right bundle branch block (RBBB), symbol (R)
- Atrial premature beats (AP), symbol (A)
- Premature ventricular beats (PVC), symbol (V)

4.6 Pre-processing

ECG data inherently carries a multitude of unwanted noise and artifact sources, including baseline drift, electrode contact noise, polarization noise, internal amplifier noise, noise from muscle activity, and motor artifacts. Noise often arises from electrode movements. In the realm of ECG denoising, common tools like bandpass filters, low-pass filters, and wavelet transforms come into play. Therefore, a crucial step involves eliminating baseline drift and the aforementioned noise to prepare ECG data for subsequent feature extraction. In this context, we propose the application of wavelet filtering to ECG signals. This method effectively computes R-peak positions without compromising the original signal's temporal position or shape. Specifically, we employed Sym4 from the Symlet wavelet function family as the chosen wavelet function for this project [19]. Additionally, we harnessed the intrinsic capability of CNN to autonomously extract features from the signal, thus enhancing network generalization while minimizing signal distortion, thus adhering to a basic filtering approach.

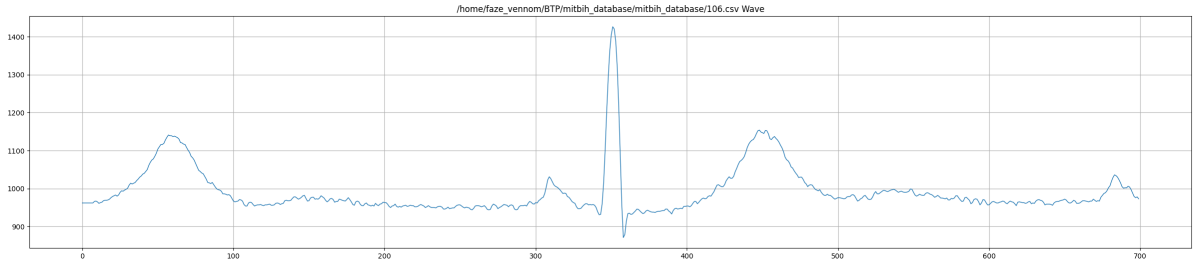

Figure 4.4: ECG signal before filtering

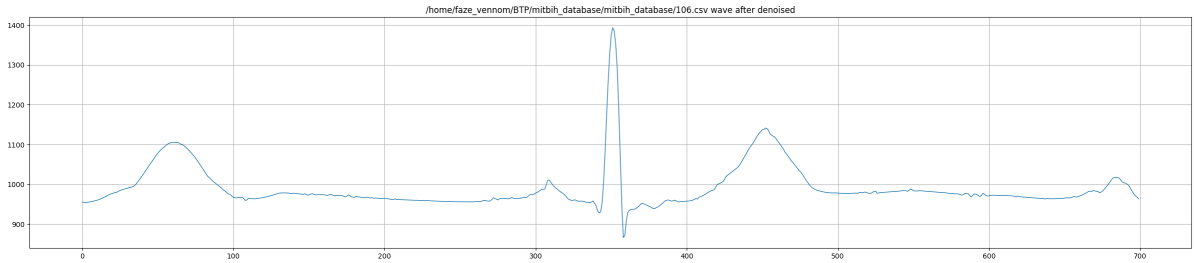

Figure 4.5: ECG signal after filtering

4.7 Data Segmentation

Within the MIT-BIH dataset, each heartbeat is meticulously annotated with its corresponding disease classification. In this study, heartbeats are methodically categorized into five distinct classes: normal, left bundle branch block, right bundle branch block, atrial premature beats, and premature ventricular beats. Notably, there is a specific classification for normal heartbeats that are premature ventricular contractions (PVC). In the initial stages of the process, the Pan-Tompkins algorithm proves invaluable for identifying the R-peaks within the ECG signals. Once these critical R-peaks are identified, the dataset, consisting of 360 samples, is centered around these reference points. Furthermore, the procedure focuses on a singular lead from the dataset. To ensure uniformity and facilitate analysis, the Z-score method is adeptly employed to normalize all signal segments.

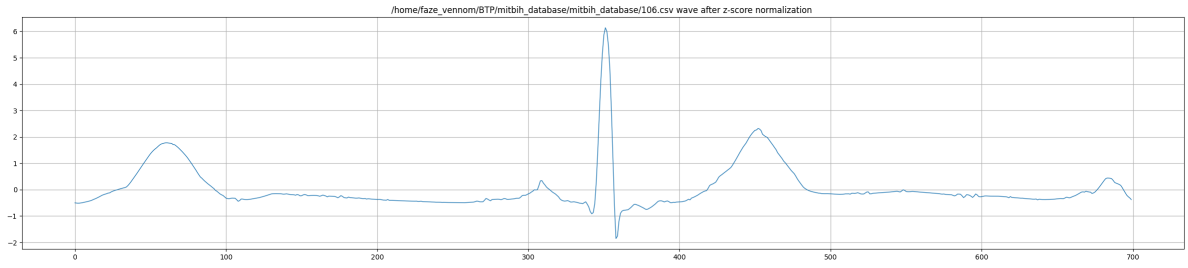

Figure 4.6: ECG signal after z-score normalization

4.8 Data Enhancement

An unbalanced dataset, where some classes have many more examples than others, can cause problems when training a model. It might make the model overly focus on the classes with more examples, which can lead to mistakes when it encounters the less rep-

resented classes.

In Table 3.2, after we cleaned up the data and split it into segments, we made sure that each category (like c2, c3, c4, and c5) had the same number of examples, specifically 15,000 each. We did this by adding more data to the categories with fewer examples and reducing the data in the category with more examples. This way, all the categories are treated equally during training, and there's no bias towards the categories with more data.

Table 4.2: Unbalanced and balanced dataset

Class	**Type**	**Unbalanced**	**Balanced**
C1	NOR	75011	15000
C2	LBBB	8071	15000
C3	RBBB	7255	15000
C4	AP	7129	15000
C5	PVC	2546	15000
		Total Data = 100012	Total Data = 75000

4.9 Understanding Hyperparameters and GlobalAverage-Pooling

How is the training of a Convolutional Neural Network (CNN) accomplished? The training process of CNNs revolves around the acquisition of filter values through a method known as backpropagation. To put it simply, a layer within a CNN is deemed "learnable" when it encompasses weight matrices. The quantity of parameters within a specific layer is determined by the number of "learnable" elements contained within its filters, essentially representing the unique parameters associated with that layer. In this context, parameters refer to weight matrices that undergo adjustments during the backpropagation procedure, thereby contributing significantly to the model's predictive abilities. The modification of these parameters is contingent upon the chosen training methodology, notably the optimization strategy implemented.

Examples of such parameters can include:

- The weights or coefficients in logistic and linear regression models.
- The weights and biases in neural networks.
- The centroids in clustering algorithms.

The formula to determine parameters in a layer is expressed as:

$$P = (K * I + 1) * F$$

Where:

P = Number of Parameters in a Layer

K = Kernel Size

I = Input Channels

F = Number of Filters

1 = Bias

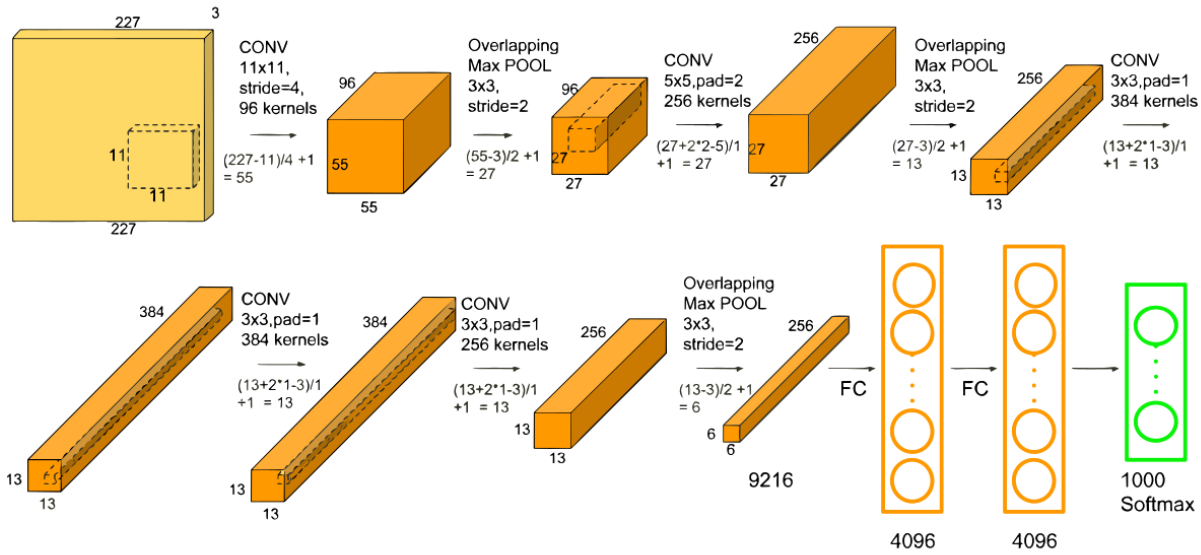

Figure 4.7: Pictorial representation of parameters and hyperparameters in a demo CNN model

Hyperparameters: While parameters are crucial to the model, it's important to note the presence of hyperparameters in the training process. Hyperparameters are configuration settings that are established before training begins and remain unchanged once training is completed. They are not considered part of the final model but play a vital role in the learning process.

Illustrations of Hyperparameters Include:

- The division ratio between training and testing datasets.
- The learning rate employed within optimization algorithms, such as gradient descent.
- The selection of an optimization algorithm, be it gradient descent, stochastic gradient descent, or the Adam optimizer.
- The designated dropout rate within NN.
- The dimensions of kernels or filters within convolutional layers.
- The size of pooling operations.
- The batch size for data processing.

Table 4.3: Details of Hyperparameters Used in 1D CNN Training

Parameter	**Value**
Size of the input layer	186×1
Activation used	ReLU
Sample size	10
Number of classes	5
Optimizer used	Adam
Regularization	0.2
Learning rate (λ)	1×10^{-3}
Batch size	30
Number of epochs	100

Optimization Algorithm: The Adam optimizer can be conceptualized as an amalgamation of two concepts: RMSprop and stochastic gradient descent with momentum [13]. Its mathematical formulation is represented as follows:

$$\upsilon_t = \beta_1 * \upsilon_{t-1} - (1-\beta_1) * g_t$$

$$s_t = \beta_2 * s_{t-1} - (1-\beta_2) * g_t^2$$

$$\Delta\omega_t = -\eta \frac{v_t}{\sqrt{s_t + \varepsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

η : Initial Learning Rate of the model

g_t : Gradient at the time instant t along ω^3

v_t : Exponential Average value of gradients along ω_j

s_t : Exponential average of the squares of gradients along ω_j

β_1, β_2 : Hyperparameters used

Global Average Pooling is an innovative technique in CNN. Instead of traditional fully connected layers, it employs a global average pooling layer to compute category confidence through spatial feature map averages from the final mlpconv layer. This vector is then directed to the softmax layer for classification. This approach fosters a transparent link between feature maps and categories by employing a micro network for local modeling, improving interpretability. Additionally, it inherently guards against overfitting, unlike fully connected layers that often rely on dropout regularization.

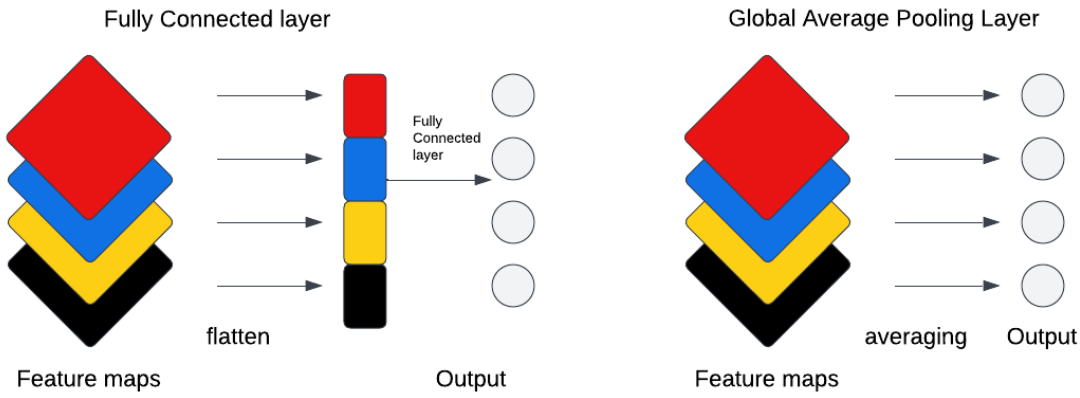

Figure 4.8: Dense layer vs Global average pooling layer

4.9.1 Loss Function

The choice of loss function and optimizer is crucial and depends on the specific characteristics of the task and dataset. Categorical Crossentropy and Adam optimizer are commonly used choices, and there is a reason they are used in this case. The man reason is the kind of operations we want to perform in this case. We want to make a multi-class classification by classifying the ECG signals into different categories.
Cost Function: The cost function quantifies the disparity between the target values and the predicted labels. An optimizer function is then utilized to minimize this disparity. While neural networks employ a range of cost functions, the cross-entropy function stands out as one of the most commonly employed.

$$C = \frac{-1}{n} \sum_{c=1}^{N} [y_c \log(a_c) + (1 - y_c) \log(1 - a_c)]$$

- C represents the cost function that is desired to be minimized.
- y_c is the target value, where c is the class index.
- N is the total number of classes.
- a is the true value.
- Gradient descent with a learning rate of 0.001 is used as the optimization method.

Categorical Crossentropy as the Loss Function: Cross-Entropy Loss, often referred to as Log Loss, serves as a prevalent loss function in machine learning, particularly in classification tasks. Its primary role is to gauge the disparity between the predicted probability distribution and the genuine distribution of target classes. In essence, it quantifies the alignment between predicted probabilities and actual labels.

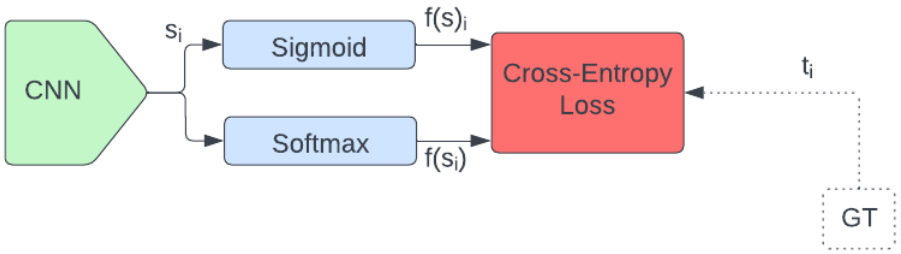

Figure 4.9: Cross-Entropy loss

$$L(y, p)=-(y \log(p)+(1-y)\log(1-p))$$

- L(y,p) is the loss between the true label y and the predicted probability p.
- y is the true label (0 or 1 in binary classification)
- p is the predicted probability that the sample belongs to class 1.

In multi-label classification, a combination of sigmoid and softmax is employed. Here, each sample can pertain to multiple classes, and the objective is to independently predict the probability of each class. This is achieved by utilizing a sigmoid activation function for each class, resulting in multiple sigmoid functions, one for each class.

$$\mathrm{f(s)}_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

$$L(y, p)=-\sum_{i=1}^{n} y_i log(p_i)$$

- $L(y, p)$ is the loss between the true label vector y and the predicted probability vector p.

- y_i is the true label for class i (1 if the sample belongs to class i, 0 otherwise).
- p_i is the predicted probability for class i.

Categorical Crossentropy emerges as a favored loss function for our endeavor, which involves categorizing ECG signals into distinct classes (such as normal or arrhythmia). Its popularity is rooted in several key advantages:

1. Suitability for Classification: Categorical Crossentropy is tailored for classification tasks where each input is assigned to precisely one category—precisely the framework we employ in ECG classification.

2. Differentiation and Backpropagation: It furnishes gradients that seamlessly facilitate the training of neural networks through backpropagation, allowing the network to iteratively fine-tune its parameters.

3. Interpretability: The loss value generated by Categorical Crossentropy is readily interpretable, rendering it an invaluable tool for monitoring model performance throughout the training process.

The ReLU activation function has been chosen for our arrhythmia classification model due to its superior performance compared to the LReLU activation function. The following mathematical equation represents the ReLU activation function:

$$ReLU(x) = max(0, x)$$

The Adam Optimizer: Our choice of the Adam optimizer is underpinned by its unique attributes and advantages: 1. Adaptivity: Adam showcases the remarkable ability to dynamically adjust the learning rate for each parameter. This adaptiveness is particularly well-suited for training neural networks grappling with gradients of varying magnitudes, potentially expediting convergence.

2. Efficiency: By maintaining individualized learning rates for each parameter and leveraging exponentially moving averages of gradients, Adam fosters stability and efficiency during training, leading to more robust and expedited convergence.

3. Ease of Use: In practice, Adam often necessitates less fine-tuning of hyperparameters compared to other optimization algorithms such as Stochastic Gradient Descent (SGD), making it a pragmatic choice for many deep learning applications.

Certainly, there are various other loss functions and optimizers you can consider for your

ECG classification project, depending on your specific requirements and the characteristics of your dataset. Here are some alternatives:

Loss Functions:

Binary Crossentropy: If your classification task is binary (e.g., classifying ECG signals as normal or abnormal), you can use binary crossentropy as the loss function instead of categorical crossentropy.

Sparse Categorical Crossentropy: If your labels are integers (class indices) instead of one-hot encoded vectors, you can use sparse categorical crossentropy.

Kullback-Leibler Divergence (KL Divergence): This is commonly used in probabilistic models and can be useful if you want to measure the difference between the predicted and actual probability distributions of classes.

Hinge Loss: Often used in support vector machines and for tasks like image classification, especially in cases where you want to maximize the margin between classes.

Focal Loss: Useful for dealing with imbalanced datasets as it down-weights well-classified examples, allowing the model to focus on hard-to-classify examples.

Dice Loss: Commonly used in medical image segmentation tasks and can be adapted for multi-class classification.

Optimizers:

Stochastic Gradient Descent (SGD): This is a fundamental optimizer that can work well with various types of neural networks. You can use it with or without momentum.

RMSprop (Root Mean Square Propagation): Similar to Adam, RMSprop adapts the learning rates per parameter, but it does not use moving averages of gradients. It can be a good alternative to Adam.

AdaGrad, short for Adaptive Gradient Algorithm, adjusts the learning rate of each parameter according to past gradient data, making it particularly suitable for handling sparse datasets.

Nadam (Nesterov-accelerated Adaptive Moment Estimation): Combines Nesterov momentum with the benefits of Adam. It's a variation of Adam that can sometimes converge faster.

Adadelta: An extension of AdaGrad that aims to alleviate its monotonically decreasing learning rate issue.

L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno): An optimizer that uses

quasi-Newton methods, which can be effective for small to medium-sized neural networks.

Proximal Gradient Descent: Particularly useful for tasks with L1 or L2 regularization, as it can enforce sparsity or weight decay.

4.10 Mechanism

1. **Input Layer**: The input layer serves a fundamental purpose by defining the dimensions of the input image. It doesn't possess any learnable attributes; hence, there are no parameters associated with it.

2. **Convolutional Layer (CONV)**: This layer serves as the core of learning within a Convolutional Neural Network (CNN). Within this layer, weight matrices play a central role. To ascertain the total number of parameters, one must account for various factors, including the filter's dimensions, both in terms of width (m) and height (n), as well as the number of filters in the preceding layer (d). Additionally, the bias term for each filter needs to be considered. Mathematically, the parameters of a CONV layer can be represented as $((m*n*d)+1)*k)$, with the '+1' accounting for the bias term associated with each filter. This equation can also be expressed as $(($width of the filter $*$ height of the filter $*$ number of filters in the preceding layer $+$ $1) *$ number of filters$)$, where "filter" denotes the quantity of filters present in the active layer.

3. **Pooling Layer (POOL)**: The pooling layer performs a specific calculation without involving any learning through backpropagation. Consequently, there are no learnable parameters associated with this layer.

4. **Fully Connected Layer (FC)**: Certainly, these layers possess the remarkable capability to acquire intricate characteristics. In fact, compared to their counterparts in other layers, this category stands out as having the most extensive parameterization. This is primarily due to the unique characteristic where every neuron establishes connections with each and every other neuron within this layer. To determine the precise count of parameters within this layer, one simply multiplies the number of neurons in the current layer (c) by the number of neurons in the preceding

layer (p), all while taking into account the essential bias term. Hence, the parameter count for this layer is calculated as $((\text{number of neurons in the current layer}, c * \text{number of neurons in the preceding layer}, p) + 1) * c$.

5. **Softmax Layer**: Much like other activation functions such as RELU, Tanh, or Sigmoid, Softmax is a parameter-free function that requires no training. It simply computes the exponential value of each logit and subsequently normalizes the resulting vector by the exponential sum.

4.11 Implementation Details

In this study, we suggested a modified 12-layer deep CNN in comparison to [6] in order to identify five micro classes included in the arrhythmia dataset while also considerably reducing the amount of trainable parameters in comparison to [6], from 292,887 to 106,997. Multiple changes were made in [6] regarding the network's structure:

1. In the first layer (CNN layer), the kernel size is changed from 13 to 9, while keeping all other parameters the same, i.e., filter size = 16, strides = 1, activation function = rectified linear activation unit (ReLU), input shape = 360.

   (a) Parameters in 12-layer deep CNN $= (13 \times 1 + 1) \times 16 = 224$

   (b) Parameters in modified 12-layer deep CNN $= (9 \times 1 + 1) \times 16 = 160$

2. In the second layer (pooling layer), pool size and strides size remain the same, i.e., 3 and 2 respectively.

   (a) Parameters in 12-layer deep CNN $= 0$

   (b) Parameters in modified 12-layer deep CNN $= 0$

3. In the third layer (CNN layer), the kernel size is reduced to 11 from 15 and filter size is reduced to 24 from 32, while keeping all other parameters the same, i.e., strides = 1, activation function = ReLU.

   (a) Parameters in 12-layer deep CNN $= (15 \times 16 + 1) \times 32 = 7712$

   (b) Parameters in modified 12-layer deep CNN $= (11 \times 16 + 1) \times 24 = 4248$

4. In the fourth layer (pooling layer), pool size and strides size remain the same, i.e., 3 and 2 respectively.

   (a) Parameters in 12-layer deep CNN $= 0$

   (b) Parameters in modified 12-layer deep CNN $= 0$

5. In the fifth layer (CNN layer), the kernel size is reduced to 13 from 17 and filter size is reduced to 56 from 64, while keeping all other parameters the same, i.e., strides = 1, activation function = ReLU.

   (a) Parameters in 12-layer deep CNN $= (17 \times 32 + 1) \times 64 = 34880$

   (b) Parameters in modified 12-layer deep CNN $= (13 \times 24 + 1) \times 56 = 17528$

6. In the sixth layer (pooling layer), pool size and strides size remain the same, i.e., 3 and 2 respectively.

   (a) Parameters in 12-layer deep CNN $= 0$

   (b) Parameters in modified 12-layer deep CNN $= 0$

7. In the seventh layer (CNN layer), the kernel size is reduced to 15 from 19 and filter size is reduced to 96 from 128, while keeping all other parameters the same, i.e., strides = 1, activation function = ReLU.

   (a) Parameters in 12-layer deep CNN $= (19 \times 64 + 1) \times 128 = 1,55,776$

   (b) Parameters in modified 12-layer deep CNN $= (15 \times 56 + 1) \times 96 = 80,736$

8. In the eighth layer (pooling layer), pool size and strides size remain the same, i.e., 3 and 2 respectively.

   (a) Parameters in 12-layer deep CNN $= 0$

   (b) Parameters in modified 12-layer deep CNN $= 0$

9. In the ninth layer, dropout is kept the same, i.e., 0.5.

   (a) Parameters in 12-layer deep CNN $= 0$

   (b) Parameters in modified 12-layer deep CNN $= 0$

10. In the tenth layer, instead of using the first Dense layer, we used a convolution layer with 5 nodes as input (i.e., filter size = 5), kernel size = 9, and activation function = ReLU.

    (a) Parameters in 12-layer deep CNN $= (35 \times 2688 + 1 \times 35) = 94,115$

    (b) Parameters in modified 12-layer deep CNN $= (9 \times 96 + 1) \times 5 = 4,325$

11. In the eleventh layer, instead of using the second Dense layer, we used a GlobalAveragePooling1D. So instead of using a dense layer, we used global average pooling.

    (a) Parameters in 12-layer deep CNN $= (5 \times 35 + 1 \times 5) = 180$

    (b) Parameters in modified 12-layer deep CNN $= 0$

12. In the twelfth layer (softmax layer).

    (a) Parameters in 12-layer deep CNN $= 0$

    (b) Parameters in modified 12-layer deep CNN $= 0$

Chapter 5

Experiment & Results

Chapter 5 is structured to encompass various aspects of our experiments and results. It commences with an overview of the evaluation metrics. We then delve into experiment design. The next section provides a detailed discussion of our results, including insights and analysis. A classification report highlights key performance indicators. Lastly, we present graphical comparisons for enhanced result visualization. This chapter offers a comprehensive exploration of our experimentation process and outcomes.

5.1 Metrics Used

1. Accuracy

   The fundamental metric frequently employed for model evaluation is commonly known as Accuracy, which quantifies the proportion of correct predictions out of the entire prediction set. This Accuracy metric is computed as:

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{FalsePositive} + \text{FalseNegative} + \text{TrueNegative}}$$

2. Precision

   Precision gives the score that represents how many times the class is predicted correctly in relation to the number of samples of the class. Precision is calculated as:

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}}$$

3. Recall

   Recall gives the score that represents how many times the class is predicted cor-

rectly in relation to the total number of times the class is predicted. Recall is calculated as:

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegatives}}$$

4. F1 Score

   The F1 score is the average of Recall and Precision but it is weighted. F1 score is calculated as:

$$\text{F1 score} = 2 * \frac{(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

5.2 Experiment Design

The model consists of a total of 12 layers, initially arranged in a structured sequence of 8 layers, alternating between convolutional and pooling layers. This arrangement encompasses 4 convolutional and 4 pooling layers. Then, a dropout layer is introduced, followed by two dense layers, culminating in a final layer with a softmax activation function. During training, the model underwent 100 epochs, with a batch size of 36.

Each convolutional layer is distinguished by several key parameters, including the number of filters, kernel size, strides, padding, and activation function. Similarly, each pooling layer specifies parameters such as the pool size and strides size. For instance, in the first layer, denoted as a convolutional layer (CNN), there are 16 filters with a kernel size of 9, and 'same' padding is applied to ensure that the output maintains the input dimensions. This layer employs the rectified linear activation function ('relu'). In contrast, the second layer, a pooling layer, utilizes a pool size of 3 and strides of 2 to facilitate the down sampling process.

This architectural pattern continues through the subsequent layers, with varying filter count, kernel size, and activation functions. Ninth layer introduces dropout with a rate of 50%, effectively deactivating half of the activation nodes to mitigate overfitting. Layer 10, another convolutional layer, is designed for 5 output classes with a kernel size of 9, single-unit strides, and 'relu' activation.

Layer 11 adopts globalaveragepooling as an alternative to dense layers. It averages feature maps, reducing the risk of overfitting. Finally, Layer 12, the softmax layer, comprises 5 inputs and predicts 5 outputs. Its primary function is to calculate the exponents of logic values and subsequently normalize the output vector based on the sum of these exponents.

5.3 Results & Discussions

From table 5.1 and 5.2, trainable parameters are reduced from 292,887 to 106,997 in modified 12-layer deep one dimensional model.

Table 5.1: Summary table of 12-layer deep one dimensional model

Model: ”sequential_5”

Layer (type)	**Output Shape**	**Param #**
conv1d_20 (Conv1D)	(None, 360, 16)	224
average_pooling1d_20 (AveragePooling1D)	(None, 179, 16)	0
conv1d_21 (Conv1D)	(None, 179, 32)	7,712
average_pooling1d_21 (AveragePooling1D)	(None, 89, 32)	0
conv1d_22 (Conv1D)	(None, 89, 64)	34,880
average_pooling1d_22 (AveragePooling1D)	(None, 44, 64)	0
conv1d_23 (Conv1D)	(None, 44, 128)	155,776
average_pooling1d_23 (AveragePooling1D)	(None, 21, 128)	0
flatten_5 (Flatten)	(None, 2688)	0
dropout_5 (Dropout)	(None, 2688)	0
dense_10 (Dense)	(None, 35)	94,115
dense_11 (Dense)	(None, 5)	180
softmax_5 (Softmax)	(None, 5)	0

Total params: 292,887 (1.12 MB)

Trainable params: 292,887 (1.12 MB)

Non-trainable params: 0 (0.00 Byte)

Table 5.2: Summary table of Modified 12-layer deep one dimensional model

Model: ”sequential”

Layer (type)	**Output Shape**	**Param #**
conv1d_20 (Conv1D)	(None, 360, 16)	160
average_pooling1d_20 (AveragePooling1D)	(None, 179, 16)	0
conv1d_21 (Conv1D)	(None, 179, 24)	4248
average_pooling1d_21 (AveragePooling1D)	(None, 89, 24)	0
conv1d_22 (Conv1D)	(None, 89, 56)	17528
average_pooling1d_22 (AveragePooling1D)	(None, 44, 56)	0
conv1d_23 (Conv1D)	(None, 44, 96)	80736
average_pooling1d_23 (AveragePooling1D)	(None, 21, 96)	0
flatten_5 (Flatten)	(None, 21, 96)	0
dense_10 (Dense)	(None, 21, 5)	4325
dense_11 (Dense)	(None, 5)	180
softmax_5 (Softmax)	(None, 5)	0

Total params: 106,997

Trainable params: 106,997

Non-trainable params: 0

5.4 Graphical Comparison

In Figures 5.1, 5.3, 5.5, 5.7 annd 5.9, we can observe a comparative analysis of the loss and epochs for proposed & existing models. This analysis encompasses both the training and test datasets. The term ”loss” serves as a quantified measure of how accurately the model predicts outcomes. A loss value of zero signifies precise model predictions, while higher loss values indicate less accurate predictions. An ”epoch” corresponds to a complete cycle of forwarding and backward propagation through the entire dataset within the neural network. To manage the computational load, we divide each epoch into smaller batches, ensuring more efficient processing.

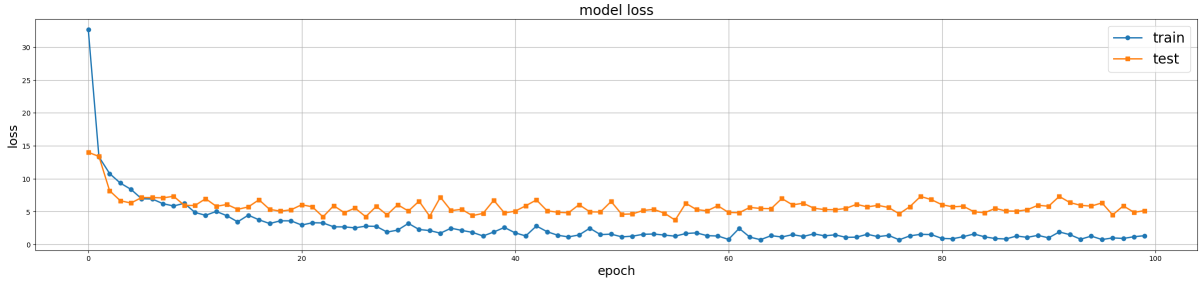

Figure 5.1: Loss v/s Epoch graph of Augmentation with CNN model to get the overall idea of loss for 100 epochs.

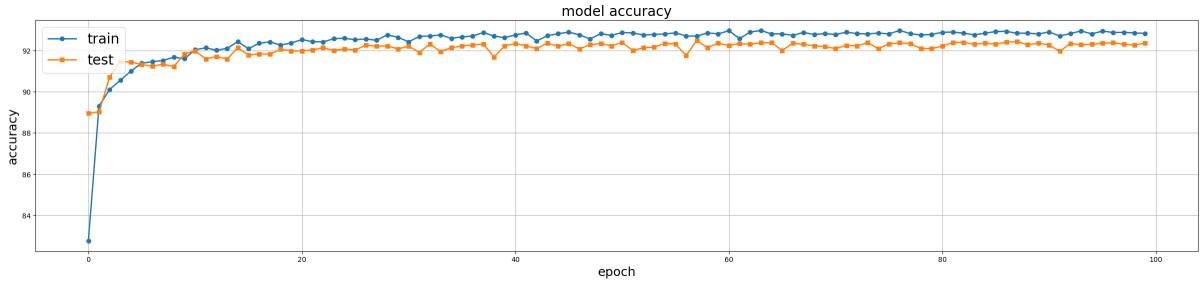

Figure 5.2: Accuracy v/s Epoch graph of Augmentation with CNN model to get the overall idea of accuracy for 100 epochs.

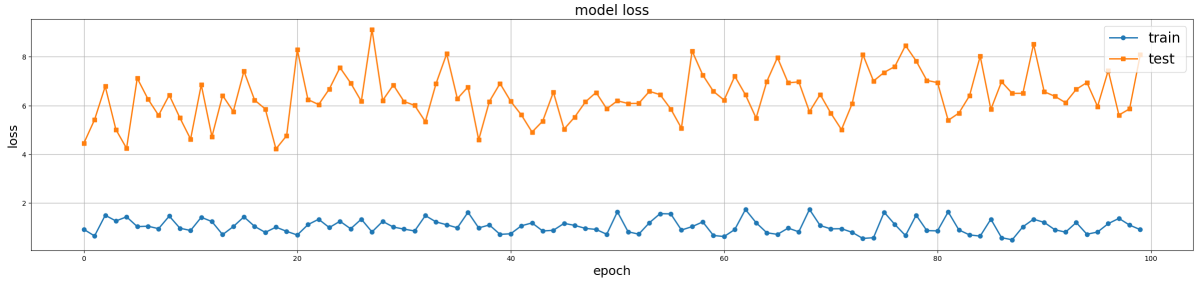

Figure 5.3: Loss v/s Epoch graph of DWT with SVM model to get the overall idea of loss for 100 epochs.

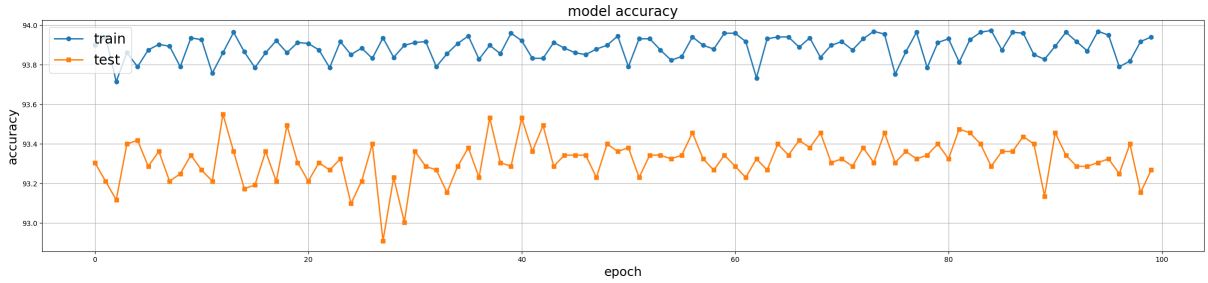

Figure 5.4: Accuracy v/s Epoch graph of DWT with SVM model to get the overall idea of accuracy for 100 epochs.

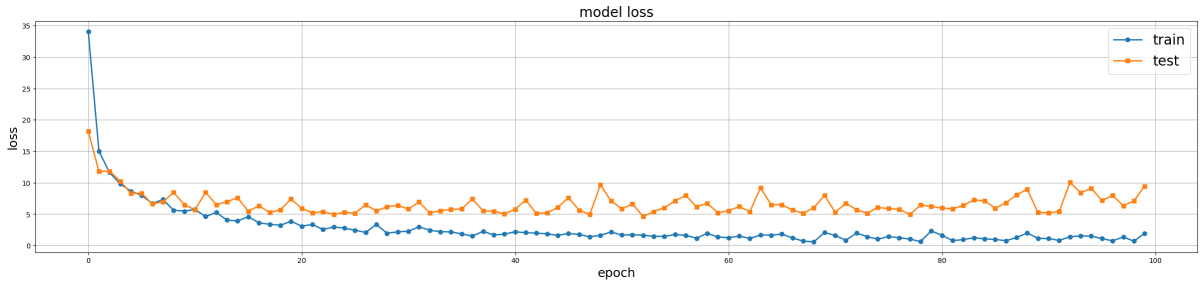

Figure 5.5: Loss v/s Epoch graph of DWT with Random forest model to get the overall idea of loss for 100 epochs.

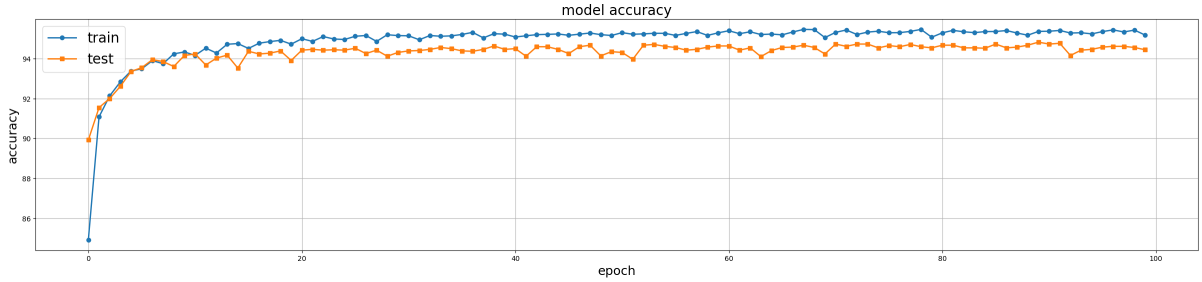

Figure 5.6: Accuracy v/s Epoch graph of DWT with random forest model to get the overall idea of accuracy for 100 epochs.

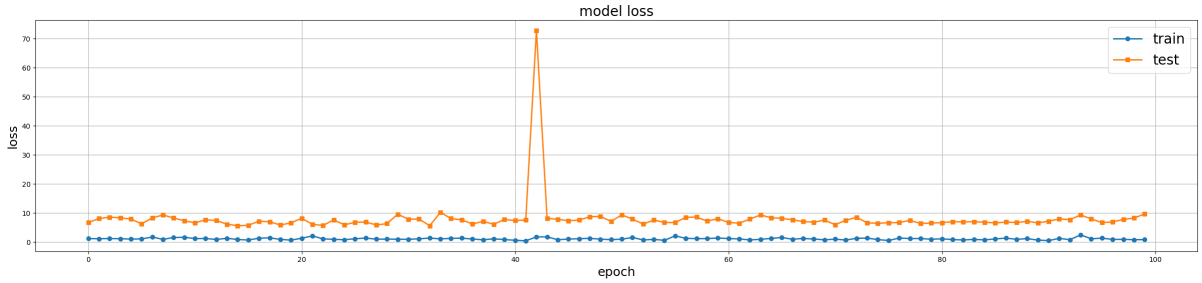

Figure 5.7: Loss v/s Epoch graph of BBNN with PSO model to get the overall idea of loss for 100 epochs.

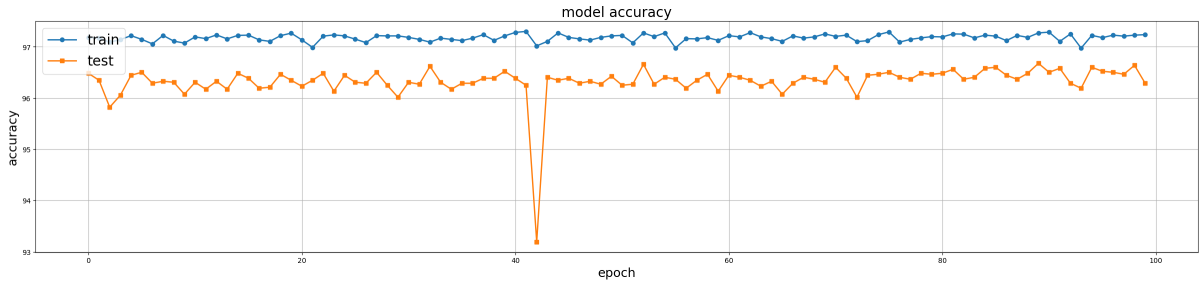

Figure 5.8: Accuracy v/s Epoch graph of BBNN with PSO model to get the overall idea of accuracy for 100 epochs.

Figure 5.2, 5.4, 5.6, 5.8 and 5.10 displays the comparison of accuracy and epoch for all the existing models and our proposed model, respectively, on both the training and test data. By calculating the percentage difference between the model predictions and the actual values, accuracy assesses how effectively our model predicts the future.

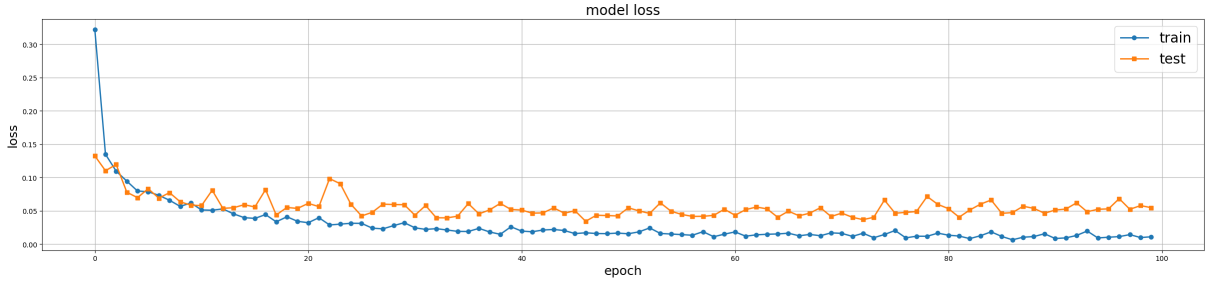

Figure 5.9: Loss v/s Epoch graph of modified 12-layer deep 1D CNN model to get the overall idea of loss occured for 100 epochs.

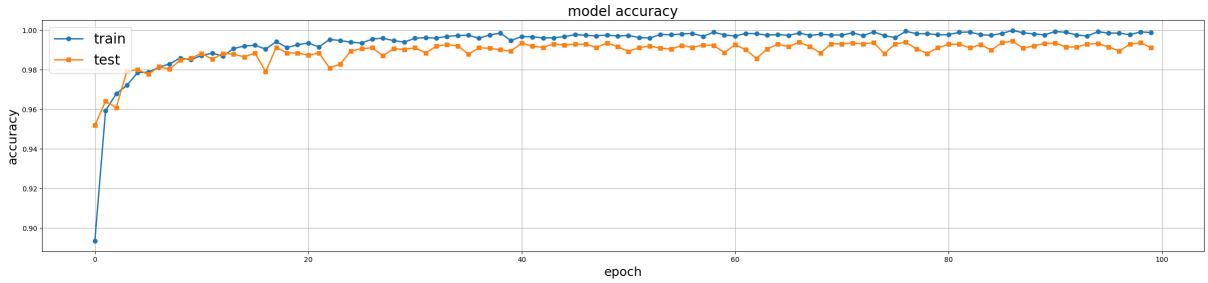

Figure 5.10: Accuracy v/s Epoch graph of modified 12-layer deep 1D CNN model to get the overall idea of accuracy for 100 epochs.

5.5 Classification Report

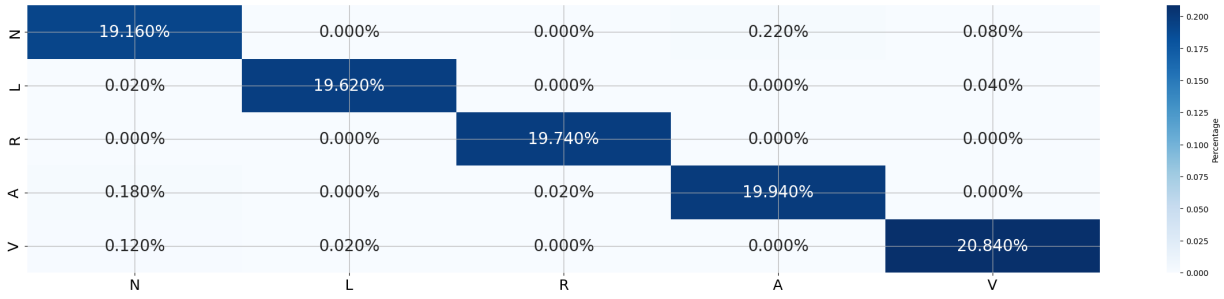

Figure 5.11: Confusion matrix of the classification model

Table 5.3: Classification report of 12-layer 1D CNN

micro-classes	Precision	Recall	F1-Score	Support
N	0.96	0.98	0.96	2997
L	0.98	0.96	0.96	2995
R	0.96	0.96	0.96	3083
A	0.95	0.96	0.95	2996
V	0.96	0.98	0.96	2929
Test Accuracy			98.16%	15000

Table 5.4: Classification report of Modified 12-layer 1D CNN

micro-classes	Precision	Recall	F1-Score	Support
N	0.99	0.98	0.99	1036
L	1.00	1.00	1.00	995
R	1.00	1.00	1.00	961
A	0.98	1.00	0.99	1000
V	0.98	0.99	1.00	1008
Test Accuracy			99.35%	5000

Chapter 6

Conclusion

This study presents a state-of-the-art deep residual CNN approach for automated detection and classification of diverse ECG heartbeat patterns, crucial for early detection of cardiovascular diseases. Our employed 1D CNN model adeptly categorizes five distinct ECG heartbeat classes, exhibiting notably superior overall accuracy compared to contemporary approaches. A pivotal feature of this work is its fully automated framework for auto-diagnosis, eliminating the need for manual feature engineering or selection at any stage. Moreover, we effectively mitigate data imbalance concerns through a resampling technique. Notably, our model achieves a markedly superior F1 score compared to comparable methods, rendering it a compelling choice for practical applications, particularly in real-time clinical auto-diagnosis scenarios grappling with imbalanced data. In the future, we contemplate the application of Long Short-Term Memory (LSTM) to further enhance the model's performance. Given the time-series nature of ECG data with enduring dependencies, LSTM represents a promising complement to CNNs.

Table 6.1: Comparison with previous works in terms of accuracy

Work	**Approach**	**Average Accuracy**
Acharya et al.[9]	Augmentation along with CNN	93.5%
Martis et al.[14]	DWT with SVM	93.8%
Li et al.[16]	DWT and random forest	94.6%
Shadmand et al.[15]	BBNN + PSO	97%
This Paper	Deep residual 1D CNN	99.35%

6.1 Contributions

In the modern era, cardiovascular diseases pose a significant health challenge. Early diagnosis of cardiac arrhythmias relies heavily on Electrocardiogram (ECG) signals. However, manual ECG interpretation is often time-consuming and demands scarce medical expertise. Current research has largely prioritized accuracy, even at the expense of intensive computational resources. The reliance on sophisticated yet costly computing equipment restricts their widespread availability in medical facilities. Our project seeks to strike a balance between accuracy and computational efficiency. To achieve this, we've employed techniques like pruning, which involves reducing the complexity of convolutional layers by adjusting filter sizes and kernel dimensions. We've also replaced fully connected layers with one-dimensional global average pooling, effectively mitigating overfitting concerns. While this approach might not entirely replace expert analysis and could result in some accuracy trade-offs, it holds promise for early emergency assistance through wearable devices like smartwatches and mobiles. Additionally, it enables future consultations with skilled ECG signal analysts. Moreover, during our investigation of the MIT-BIH Arrhythmia database, we have focused on the classification of distinct subcategories. These include Normal rhythm, Left Bundle Branch Block, Right Bundle Branch Block, Atrial Premature Beats, and Premature Ventricular Beats, which is a departure from the broader five primary classes. These endeavors augment our capacity to identify subtle cardiac irregularities.

6.2 Limitations

More reduction in filters and kernel size will decrease the trainable parameters, but there may be the chances of model behaving abruptly, i.e accuracy may see a significant drop. So reduction in trainable parameters must be done cautiously.

6.3 Future Scope

In the ever-evolving landscape of micro ECG classification, future models can embrace simplicity while enhancing efficiency. Techniques like knowledge distillation can simplify models by transferring insights from larger, complex models. Additionally, quan-

tization, which reduces the bit precision of weights, can further streamline deep neural networks. Combining these strategies with innovative directions, such as real-time monitoring with wearable devices, personalized health insights, and edge computing optimization, can propel the field of cardiac arrhythmia detection towards more accessible and efficient solutions. These advancements not only broaden the project's scope but also contribute to the future of cardiac care.

Furthermore, considering ethical concerns and privacy, along with international collaboration and clinical validation, will be essential steps in ensuring the responsible and effective deployment of these technologies in real-world healthcare settings.

Bibliography

[1] K. Mc Namara, H. Alzubaidi, and J. K. Jackson, "Cardiovascular disease as a leading cause of death: how are pharmacists getting involved?" Integrated pharmacy research & practice, vol. 8, p. 1, 2019.

[2] W. Caesarendra, T. A. Hishamuddin, D. T. C. Lai, A. Husaini, L. Nurhasanah, A. Glowacz, and G. A. F. Alfarisy, "An embedded system using convolutional neural network model for online and real-time ecg signal classification and prediction," Diagnostics, vol. 12, no. 4, p. 795, 2022.

[3] S.-L. Guo, L.-N. Han, H.-W. Liu, Q.-J. Si, D.-F. Kong, and F.-S. Guo, "The future of remote ecg monitoring systems," Journal of geriatric cardiology: JGC, vol. 13, no. 6, p. 528, 2016.

[4] W. Yin, X. Yang, L. Zhang, and E. Oki, "Ecg monitoring system integrated with ir-uwb radar based on cnn," IEEE Access, vol. 4, pp. 6344–6351, 2016.

[5] S.-N. Yu and Y.-H. Chen, "Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network," Pattern Recognition Letters, vol. 28, no. 10, pp. 1142–1150, 2007.

[6] M. Wu, Y. Lu, W. Yang, and S. Y. Wong, "A study on arrhythmia via ecg signal classification using the convolutional neural network," Frontiers in Computational Neuroscience, vol. 14, 2021, doi: 10.3389/fncom.2020.564015. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncom.2020.564015

[7] S. Y. Wong, K. S. Yap, H. J. Yap, and S. C. Tan, "A truly online learning algorithm using hybrid fuzzy artmap and online extreme learning machine for pattern classification," Neural Processing Letters, vol. 42, no. 3, pp. 585–602, 2015.

[8] S. Y. Wong, K. S. Yap, H. J. Yap, S. C. Tan, and S. W. Chang, "On equivalence of fis and elm for interpretable rule-based knowledge representation," IEEE transactions on neural networks and learning systems, vol. 26, no. 7, pp. 1417–1430, 2014.

[9] U.R. Acharya, S.L. Oh, Y. Hagiwara, J.H. Tan, M. Adam, A. Gertych, and R. San Tan, "A deep convolutional neural network model to classify heartbeats," Computers in biology and medicine, vol. 89, pp. 389–396, 2017.

[10] P. Pławiak and U. R. Acharya, "Novel deep genetic ensemble of classifiers for arrhythmia detection using ecg signals," Neural Computing and Applications, vol. 32, no. 15, pp. 11 137–11 161, 2020.

[11] M. Zubair, J. Kim, and C. Yoon, "An automated ecg beat classification system using convolutional neural networks," in 2016 6th international conference on IT convergence and security (ICITCS). IEEE, 2016, pp. 1–5.

[12] U. R. Acharya, H. Fujita, S. L. Oh, Y. Hagiwara, J. H. Tan, and M. Adam, "Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals," Information Sciences, vol. 415, pp. 190–198, 2017.

[13] D. K. Atal and M. Singh, "Arrhythmia classification with ecg signals based on the optimization-enabled deep convolutional neural network," Computer Methods and Programs in Biomedicine, vol. 196, p. 105607, 2020.

[14] R. J. Martis, U. R. Acharya, C. M. Lim, and J. S. Suri, "Characterization of ECG beats from cardiac arrhythmia using discrete cosine transform in PCA framework," Knowledge-Based Syst., vol. 45, pp. 76–82, 2013.

[15] Shadmand, Shirin Mashoufi, Behbood. (2016). A new personalized ECG signal classification algorithm using Block-based Neural Network and Particle Swarm Optimization. Biomedical Signal Processing and Control. 25. 12-23. 10.1016/j.bspc.2015.10.008.

[16] T. Li and M. Zhou, "Ecg classification using wavelet packet entropy and random forests," Entropy, vol. 18, no. 8, p. 285, 2016.

[17] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," CoRR, vol. abs/1207.0580, 2012, cite arxiv:1207.0580. [Online]. Available: http://arxiv.org/abs/1207.0580

[18] M. Lin, Q. Chen, and S. Yan, "Network in network," 12 2013.

[19] B. N. Singh and A. K. Tiwari, "Optimal selection of wavelet basis function applied to ecg signal denoising," Digital signal processing, vol. 16, no. 3, pp. 275–287, 2006.

[20] S. Y. Wong, K. S. Yap, and H. J. Yap, "A constrained optimization based extreme learning machine for noisy data regression," Neuro-computing, vol. 171, pp. 1431–1443, 2016.

[21] X. Xu and H. Liu, "Ecg heartbeat classification using convolutional neural networks," IEEE Access, vol. PP, pp. 1–1, 01 2020, doi: 10.1109/ACCESS.2020.2964749

[22] T. Wang, C. Lu, Y. Sun, M. Yang, C. Liu, and C. Ou, "Automatic ecg classification using continuous wavelet transform and convolutional neural network," Entropy, vol. 23, p. 119, 01 2021, doi: 10.3390/e23010119

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, p. 2012.