Accurate & Energy-efficient ECG Classification using RRAM based DNN Architecture

by

Sudeshna Dash

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday, 19 July 2021 at 11:00

Student number: 5009944
Project duration: 1 December 2020 – 19 July 2021
Thesis committee:

Dr. ir. Rajendra Bishnoi,	CE, TU Delft, supervisor
Prof. dr. ir. Said Hamdioui,	CE, TU Delft
Dr. ir. Matin Jafarin,	DCSC, TU Delft

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Abstract

Cardiac arrhythmia characterized by irregular heartbeats is a prevalent problem among people suffering from cardiovascular diseases (CVD). Abnormalities in the heartbeats manifested in the electrocardiogram (ECG) signal are traditionally analysed by expert cardiologists or semi-automated computer aided techniques, which can be time consuming. Recent deep neural network (DNN) methods have paved the way for fully-automated timely classification. However, low accuracy and high energy consumption are few of the common problems with the existing DNN methods. In this work, a complete solution based on hierarchical three stage DNN architecture is proposed using Resistive Random Access Memory (RRAM) based Computation in-memory (CIM) architecture. The proposed scheme using Fully Connected Network (FCN), Long Short Term Memory Network (LSTM), assisted with dimensionality reduction and online personalized partial training enables a high accuracy and energy-efficient ECG arrhythmia classification solution. Experimental study conducted with MIT-BIH arrhythmia database show that the proposed scheme achieves an accuracy of 98.73%, F1 score of 97.6%, and consumes 442.25 μJ of energy. Compared to state-of-the-art, energy improvement of up to 2.83x is achieved, with a reasonable area overhead.

Acknowledgement

I would like to thank Prof. Rajendra Bishnoi, my supervising professor for his constant support and guidance throughout the course of this project. I learnt a lot from him in improving my work and getting clarity to steer the thesis in the right direction. I would also like to thank Prof. Said Hamdioui for his advice and inputs on ways to improve the thesis work. I extend my thanks to Sumit Diware for helping me with concepts, clarifying doubts and assisting me on numerous occasions. I also thank Anteneh Gebregiorgis for sharing ideas and helping me in the learning process. Their advice and scientific approach has helped in improving my skills for accomplishing this task.

A special thanks to my dearest brothers, Sandeepan and Swostik, for always being by my side through the thick and thin and motivating me to move ahead. I would never forget my friends as well. Thanks for sharing so many beautiful moments of laughter and memories that I will cherish forever. I am lucky to have you all in my life.

Last but not the least, I am forever grateful to my parents for believing in me in whatever I do. I have always hoped that I have made you proud. Thank you for your unconditional love and support. Without you none of this would have been possible.

Thank you!

Sudeshna Dash
Delft, July 2021

Contents

Abstract iii

Acknowledgement v

List of Figures xi

List of Tables xv

1 Introduction 1

1.1 Motivation . . . 1

1.1.1 Need for Deep Neural Networks (DNN) for arrhythmia classification . . 1

1.1.2 Beyond von-Neumann architectures for DNN implementation . . . 2

1.2 Problem Statement . . . 4

1.3 Contributions . . . 4

1.4 Organization . . . 5

2 Background 7

2.1 Basics of ECG . . . 7

2.1.1 ECG Arrhythmia . . . 8

2.1.2 Classification of ECG Arrhythmia . . . 10

2.1.3 Pre-Processing . . . 10

2.1.4 Training and Validation . . . 12

2.2 Basics of Deep Neural Network Topologies . . . 12

2.2.1 Fully Connected Network . . . 12

2.2.2 Long-Short-Term-Memory Networks (LSTM) . . . 13

2.2.3 Bidirectional Long-Short-Term-Memory Network (BLSTM) . . . 14

2.2.4 Temporal Convolutional Network (TCN) . . . 15

2.2.5 Hybrid Neural Networks . . . 16

2.3 Dimensionality Reduction Techniques . . . 17

2.3.1 Principal Component Analysis (PCA) . . . 17

2.3.2 Linear Discriminant Analysis (LDA) . . . 18

2.3.3 Discrete Wavelet Transform (DWT) 19
2.4 Computation in-memory using RRAM devices 20
2.4.1 Memristor Operation 21
2.4.2 RRAM devices as crossbar array 22
2.5 Design Metrics 25

3 Related Works **27**
3.1 Traditional Classification Techniques for ECG Arrhythmia 27
3.2 Convolutional Neural Network (CNN) 28
3.3 Long Short Term Memory Network (LSTM) and Bidirectional LSTM 29
3.4 Hybrid of CNN and LSTM 31
3.5 Temporal Convolutional Network (TCN) 33
3.6 Spiking Neural Network (SNN) 34
3.7 Summary 36

4 Proposed Scheme **39**
4.1 Proposed ECG Classification Scheme 39
4.2 Model of Hierarchical Three Stage DNN 40
4.3 RRAM based DNN framework 42
4.4 Exploration of DNN Topologies 42
4.5 Optimization using Dimensionality Reduction 44
4.6 Optimization using online Partial Training 46
4.7 Complete Hierarchical Three Stage DNN Architecture 46

5 Simulation Results **49**
5.1 Simulation Setup 49
5.1.1 Dataset 49
5.1.2 Software Setup 49
5.1.3 Hardware Setup 50
5.2 Exploration of DNN Topologies 50
5.3 Optimization using Dimensionality Reduction 51
5.4 Hierarchical Three Stage DNN Architecture 52
5.5 Optimization using online Partial Training 54
5.6 Comparison with state-of-the-art 54

6 Conclusion **59**

6.1 Conclusion 59
6.2 Future Works 59

A Appendix **61**

Bibliography **71**

List of Figures

1.1 Global causes of death based on data from WHO, 2017 [26] 2

1.2 Increasing processor-memory gap in von Neumann architectures [29] 3

2.1 An ECG beat represented on Voltage and Time axis consisting of 'PQRST' cardiac cycle 7

2.2 Fifteen types of ECG arrhythmia based on MIT-BIH arrhythmia database [40] . 9

2.3 Classification of ECG arrhythmia using Deep Neural Network (DNN) starting from ECG signal acquisition to identifying the output class 10

2.4 Processing steps of 'QRS' detection using Pan Tompkins algorithm 11

2.5 Fully Connected Network 13

2.6 LSTM cell composed of forget gate f_t, input gate i_t, and output gate o_t 13

2.7 An illustration of LSTM network for classification 14

2.8 An illustration of BLSTM network for classification 15

2.9 An illustration of TCN network for classification 16

2.10 An illustration of hybrid architecture combining TCN and LSTM 17

2.11 Decreasing variance ratio of principal components computed for dimensionality reduction of ECG beat. After n=15, the variance contribution of each principal component becomes close to zero 18

2.12 A four level ECG beat decomposition into Approximation (A) and Detailed (D) coefficients for performing DWT 19

2.13 Low Pass and High Pass decomposition filter for db2 wavelet 20

2.14 von Neumann bottleneck in conventional computing overcome by Computation In-Memory (CIM) architectures [17] 20

2.15 Four fundamental two-terminal circuit elements relating Voltage (V), Charge (q), Current (i) and Flux (ϕ) [2] 21

2.16 Schematic of a metal oxide RRAM based on metal-insulator-metal structure [25] 22

2.17 Characteristic IV curve of a RRAM device consisting of SET and RESET processes [45] 23

2.18 RRAM based DNN framework consisting of RRAM crossbar array and periphery devices, similar to ISAAC [51] 24

2.19 1T1R and 1S1R cell [46] 24

2.20 Confusion matrix for a binary classification task 25

3.1 Network architecture used in [35] 28

3.2 Two Stage Neural Network as proposed in [56] 29

3.3 Hardware implementation of the two layers memristor crossbar array used in [28] 29
3.4 Extreme Learning Machine based on Local Receptive Field BSLTM architecture used in [47] 30
3.5 Structure of the unidirectional and bidirectional LSTM network used in [60] . . . 31
3.6 Network architecture as a combination of CNN and LSTM used in [63] 32
3.7 Convolutional Auto Encoders with LSTM for classification used in [62] 33
3.8 ECG_TCN network architecture used in [37] 33
3.9 An illustration of a Leaky Integrate and Fire Neuron [32] 34
3.10 Network architecture proposed in [9] 35
4.1 Overview of the proposed ECG classification scheme 39
4.2 Architecture development steps 40
4.3 Schematic of Hierarchical Three Stage DNN architecture to classify input ECG beats at three different hierarchical levels 41
4.4 Stage 1 DNN topology utilising a resource efficient fully connected network to ensure less energy consumption 42
4.5 DNN topologies (a) LSTM (b) BLSTM (c) TCN (d) Hybrid of TCN & LSTM . . . 43
4.6 Qualitative comparison of various DNN topologies (please refer Section 5.2 for detailed results analysis). BLSTM has highest accuracy/F1 while LSTM has the most optimal performance. 44
4.7 Impact of dimensionality reduction on network parameters 45
4.8 Qualitative comparison of various dimensionality reduction techniques (please refer Section 5.3 for a detailed analysis). PCA with highest accuracy/F1 has the best performance. 45
4.9 Training for personalization. $*$Full network training is a standard patient-specific training choice [9]. 46
4.10 Complete Hierarchical Three Stage DNN Architecture to classify input ECG beats into hierarchical output classes. Uses dimensionality reduction for energy optimization and partial online training for accuracy optimization. 47
5.1 Performance comparison of Long Short term Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN), and Hybrid of TCN & LSTM. LSTM is the most optimized DNN topology. 52
5.2 Impact of dimensionality reduction techniques. PCA is the most optimized technique maintaining high accuracy and F1 score. 53
5.3 Confusion Matrix for Stage 1 representing number of beats classified into Normal (N) and Abnormal classes based on ‘Actual’ class and ‘Classified’ class . . 54
5.4 Confusion Matrix for Stage 2 representing number of beats classified into Supraventricular Ectopic Beat (SVEB), Ventricular Ectopic Beat (VEB), Fusion Beat (F) and Unknown Beat (Q) based on ‘Actual’ class and ‘Classified’ class 54

5.5 Stage 3 Confusion Matrix representing number of beats classified into Atrial Premature Beat (A), Aberrated Atrial Premature Beat (a), Nodal (junctional) Premature Beat (J), Supraventricular Premature Beat (S), Premature Ventricular Contraction (V), Ventricular Escape Beat (E) based on 'Actual' class and 'Classified' class 55

5.6 Variation of Accuracy and F1 Score for different patients with and without partial training 56

5.7 Energy improvement up to 2.83x compared to two stage neural network (TSNN) [56] 56

A.1 MATLAB Simulation of QRS Peak Detection using Pan Tompkins algorithm for Patient ID 100 61

A.2 A detailed representation of inside 2 layer LSTM Network 62

A.3 A 2 layer LSTM Network Structure 63

A.4 A 2 layer BLSTM Network Structure 64

A.5 A 2 layer TCN structure 66

A.6 Hybrid Network Structure 68

A.7 Stage 1 crossbar array 70

List of Tables

2.1 Types of ECG arrhythmia based on MIT-BIH database [40] 8

4.1 Purpose and energy-efficiency of each stage of the heirarchical three stage DNN model 41

5.1 Distribution of ECG Beats across each arrhythmia class 50

5.2 Configuration Parameters 50

5.3 Hardware Parameters 51

5.4 Performance of DNN Topologies 51

5.5 Impact of Dimensionality Reduction 52

5.6 Hierarchical Three Stage DNN Performance 53

5.7 Beats processed by each stage 53

5.8 Impact of partial training on Accuracy and F1 score for personalized system . 55

5.9 Comparison with state-of-the-art 57

A.1 LSTM Energy Consumption 63

A.2 FC LSTM Energy Consumption 63

A.3 LSTM Area Calculation 64

A.4 BLSTM Energy Consumption 64

A.5 FC BLSTM Energy Consumption 65

A.6 BLSTM Area Calculation 65

A.7 TCN Energy Consumption, Layer 1 67

A.8 TCN Energy Consumption, Layer >=2 67

A.9 FC TCN Energy Consumption 67

A.10 TCN Area Calculation 67

A.11 Hybrid Layer 1 Energy Consumption 69

A.12 Hybrid layer 2 Energy Consumption 69

A.13 FC Hybrid Energy Consumption 69

A.14 Hybrid architecture Area calculation 69

A.15 Stage 1 Energy Consumption 70

A.16 Stage 1 Area Calculation 70

1 Introduction

This chapter introduces the thesis. First, the motivation of the thesis is discussed, followed by the problem statement. Then, the thesis contribution is presented and finally the thesis organization is briefed.

1.1. Motivation

Cardiovascular diseases (CVDs) are a group of disorders involving the heart and the blood vessels such as heart stroke, cardiac arrest, congenital heart disease, cardiac arrhythmias etc. CVDs are reported to be the number one cause of death worldwide in 2016 ([22], [7]) as shown in Figure 1.1. According to the World Health Organization (WHO) report, it is estimated that by 2030, the number of deaths due to CVDs will reach upto 23 million. The main risk factors for CVDs include unhealthy lifestyle, hypertension, blood glucose level etc. Most CVD patients are accompanied by cardiac arrhythmia which is one of the most common problems in cardiology. Cardiac arrhythmias are slowed, accelerated or irregular heartbeats caused due to improper functioning of the heart. They are usually identified by using electrocardiogram or what is more commonly known as ECG. ECG is a physiological signal that records the electrical activity of the heart. It is widely used by medical professionals in diagnosing abnormal heart rhythms and investigating problems with the electrical conduction of the heart. Besides, early detection of cardiac arrhythmia can help prevent the emergence of any catastrophic heart diseases. Therefore, careful monitoring and identification of the arrhythmia can help mitigate the risk of loss of life.

1.1.1. Need for Deep Neural Networks (DNN) for arrhythmia classification

Cardiac arrhythmia treatment suffers from the challenge of accurate diagnosis methods. It is especially due to the inconsistency in a patient's heart signal with varying physiological conditions and heterogeneity among different patients' heart signal. Cardiologists need to manually examine long durations of ECG recordings to identify the abnormal beats. This task can become time-consuming, tedious, subjective, and costly. Sometimes the diagnosis is performed many days after the recording due to heavy workload on the doctors. This makes it difficult for the patients to know their heart problems promptly and may lead to a miss or delay in the medical treatment.

With the emergence of automatic computer-based ECG classification solutions, the above problems with manual investigation is solved. Pattern recognition techniques such as random forest [34], support vector machine [55], naive bayes etc. is applied to ECG signal for automatic classification. However, these traditional artificial intelligence (AI) techniques require manual feature extraction of ECG signals to be done before processing it through the

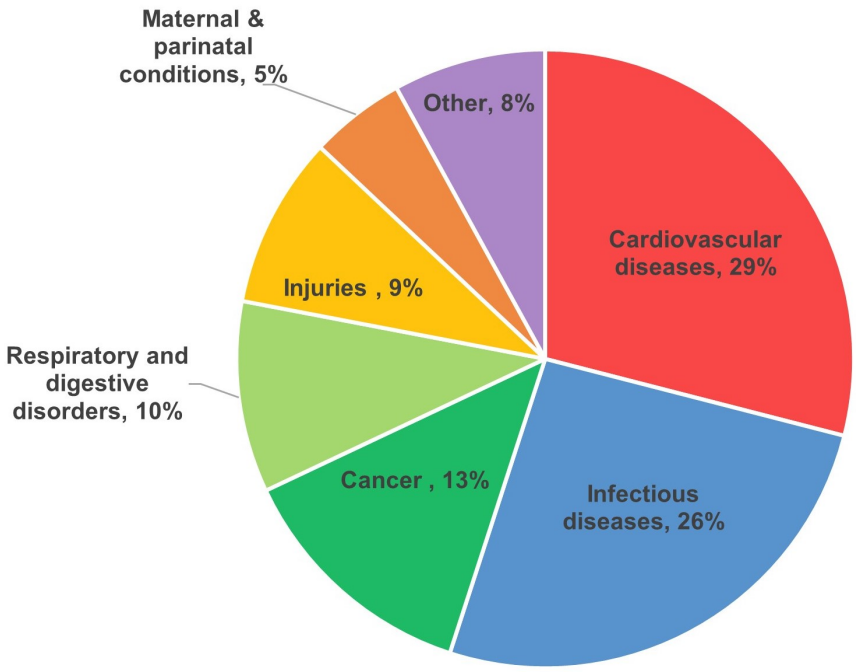

Figure 1.1: Global causes of death based on data from WHO, 2017 [26]

AI model. Besides, if the feature extraction is not precise, it will eventually have a negative impact on the classification results.

In recent years, deep learning using Deep Neural Networks (DNN) has solved this problem. Deep learning algorithms have the skill to learn features automatically, eliminating the need for manual feature extraction of inputs. It can automatically extract the useful information from the inputs, enabling accurate health-data interpretation and a timely clinical action. Several DNN architectures have been used for ECG arrhythmia classification, such as Convolutional Neural Network (CNN) ([56], [42]), Temporal Convolutional Network (TCN) [59], Long Short Term Memory Network (LSTM) ([14], [61]), Hybrid of CNN & LSTM [63], Spiking Neural Network (SNN) ([12], [16]) etc. Few of these DNN solutions are implemented in hardware ([56], [10], [12], [16]). However, many of the existing DNN solutions face the problem of low accuracy ([28], [42], [14]). Besides, only a subset of arrhythmia classes have been considered in most of the approaches ([28], [61], [12]). This is not a suitable method for practical usage and does not validate the performance of the classifier to its true potential. Hardware level solutions have either focused on achieving high accuracy [56] or low energy ([12], [16]). There is a lack of fulfilling both the requirements simultaneously. Besides, there is a need for the DNN solutions based on traditional computer architectures and memory technologies to be replaced with more advanced state-of-the-art techniques.

1.1.2. Beyond von-Neumann architectures for DNN implementation

Modern computer architectures are based on von Neumann architecture which separates the central processor unit (CPU) used for computations, from the memory unit used for storage. Data processing in von Neumann architectures requires transfer of data between the memory and the CPU through a transmission interface. With the increase in the amount of data in today's applications, the data transfer rate has reached a bottleneck. Even though our processors are becoming faster, the computation limitation arises due to performance gap between the memory and processor which has grown over the years, as shown in Figure 1.2. It is known

as the memory wall problem. This has imposed critical requirements for the performance of modern computing systems such as CPUs, Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) which are becoming inefficient for such applications. To overcome the above problem, beyond traditional approaches are required.

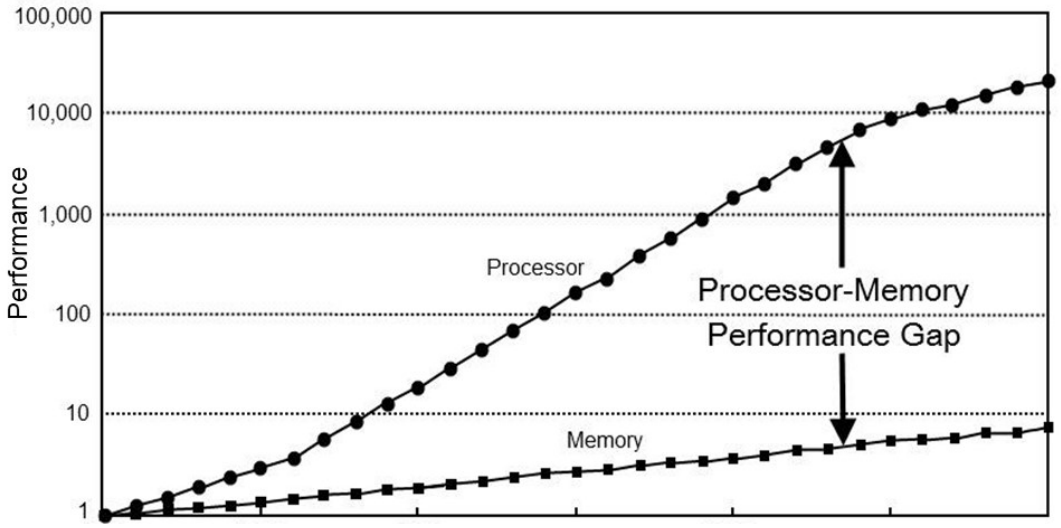

Figure 1.2: Increasing processor-memory gap in von Neumann architectures [29]

Computation in-memory (CIM) allows data storage and computation at the same location. It fuses the processing and storage unit together overcoming the data movement transfer bottleneck between memory and the processor. CIM architecture uses crossbar arrays composed of memory units for storage and follow simple circuit laws such as Ohms law and Kirchoff;s current law for computation. It has shown promising potential for a wide range of complex and cognitive tasks like visual/audio recognition, self-driving etc. [25].

Design of CIM architectures for DNNs are conventionally based on complementary metal–oxide–semiconductor (CMOS) memory technologies. These are charge based memory units which includes static random access memory (SRAM), dynamic random access memory (DRAM) and flash memory. But these CMOS based memories are facing challenges due to scaling down to 10nm and beyond. This results in large static power loss, current leakage, higher manufacturing cost, low yield, reliability and scaling issues [24]. Therefore, in recent times, various emerging memory technologies are being actively investigated as a suitable alternative to fulfill the high demand for performance. These emerging memory technologies depend upon the change of resistance to store the information. Among all the emerging memory technology, resistive random access memory (RRAM) has significant advantages such as easy fabrication, simple metal-insulator-metal (MIM) structure, excellent scalability, high speed, near-zero leakage, long data retention, and compatibility with the current CMOS technology [24].

RRAM based CIM solution for ECG classification has been adopted in [28], [12], [16]. But these works have reported low accuracy. Besides, mapping of high accuracy DNN architecture such as CNN [61] or CNN + LSTMs [63] onto RRAM based CIM architectures for hardware implementations leads to high energy consumption, due to the structure of the network. Therefore, in this thesis, RRAM based CIM architecture has been used for DNN implementation of ECG arrhythmia classification to achieve high accuracy as well as energy efficiency.

1.2. Problem Statement

- Medical data is significant and sensitive. Accurate health data interpretation can facilitate providing adequate treatment to the patient. It is also highly crucial for the solution to be able to identify all types of arrhythmia for practical usage. Therefore, performing high accuracy classification for all ECG arrhythmia classes, is elemental for performing a correct clinical action.

- Since, wearable medical devices are dependent upon batteries, frequent recharge or replacement of batteries should be avoided as much as possible. Using high-energy consuming neural network architecture for arrhythmia classification is not suitable for practical usage. Besides, since modern von Neumann computing systems based on CMOS designs are becoming inefficient for neural network applications, adopting RRAM based CIM architecture as an effective alternative for ECG classification is needed.

- Personalization of devices provides accurate experiences for individual users and makes the solution more scalable in the long run. This is especially the case for medical signals such as ECG, which is unique for a person. Therefore, developing innovative solutions to ensure high accuracy for personalization of devices is required.

1.3. Contributions

A complete solution for all classes of ECG targeting high accuracy and energy-efficiency using RRAM based DNN framework is proposed in this thesis. The key contributions of this thesis are listed below:

- **Hierarchical Three Stage DNN Architecture**
  Traditional ECG classification approaches use a single stage DNN model for identifying the arrhythmia type. In this thesis, a heirarchical three stage DNN architecture has been proposed to obtain high accuracy and energy-efficiency using RRAM based DNN framework.

- **Performance Evaluation of DNN Topologies**
  To select DNN topologies for the hierarchical three stage DNN architecture, a comprehensive evaluation of state-of-the-art DNN topologies, including Long-Short term Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN), and hybrid of TCN & LSTM was done. Based on accuracy and energy trade-off, it was concluded that LSTM is the most suitable topology for the hierarchical three stage DNN model.

- **Optimization for energy-efficiency and accuracy**
  A method for optimizing neural network models for energy-efficiency and accuracy was done with dimensionality reduction and personalized partial training, respectively. To reduce energy consumption, several existing dimensionality reduction techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Discrete Wavelet Transform (DWT) were explored and analysed for their accuracy and energy trade-off. It was concluded that PCA is the most suitable dimensionality reduction technique for the application. To regain accuracy for new individual users, personalized training of the hierarchical three stage DNN architecture was done using energy-efficient partial training approach.

The proposed scheme obtains a high accuracy of 98.73%, and F1 score of 97.6%. Compared to state-of-the-art two stage network scheme [56], energy improvement of up to 2.83x

is obtained with the proposed solution. Due to the three stage topology, the solution has a reasonable area overhead.

1.4. Organization

The rest of the thesis is organized as follows:

- Chapter 2 of the thesis gives a background on the basic concepts utilized in the thesis. It discusses the basics of ECG classification, basics of neural network models, dimensionality reduction techniques and provides understanding about RRAM based CIM architecture.
- Chapter 3 presents the state-of-art work on ECG classification using DNNs. Key contributions and the drawbacks of the works are discussed.
- Chapter 4 gives details about the proposed scheme of ECG classification. Hierarchical three stage DNN architecture, optimization with dimensionality reduction techniques and partial training for the personalization are presented in this chapter.
- Chapter 5 describes the simulation results. The simulation setup of the work is presented, followed by the experimental results and detailed analysis of the proposed ECG classification scheme.
- Chapter 6 concludes the thesis. Few suggestions for possible future works is also given in this chapter.

2

Background

This chapter gives a background of the thesis work. Basics of ECG signal and types of ECG arrhythmia are discussed in detail in Section 2.1. The basics of DNN topologies and their working is presented in Section 2.2. Then, RRAM based CIM architectures and working of the Resistive RAM (RRAM) devices, which forms the building block of these CIM architectures, is given in Section 2.4. Section 2.5 discusses the design metrics considered in the thesis.

2.1. Basics of ECG

The electrocardiogram signal, also known as ECG signal is a universal characteristic of the human heart. It has been used for several decades as an efficient and reliable diagnostic tool in medical applications. It is a graph of voltage versus time to demonstrate the electrical activity of the heart. The ECG signal tracks the rhythm of the heart, and changes in the normal ECG pattern indicates the occurrence of numerous cardiac abnormalities.

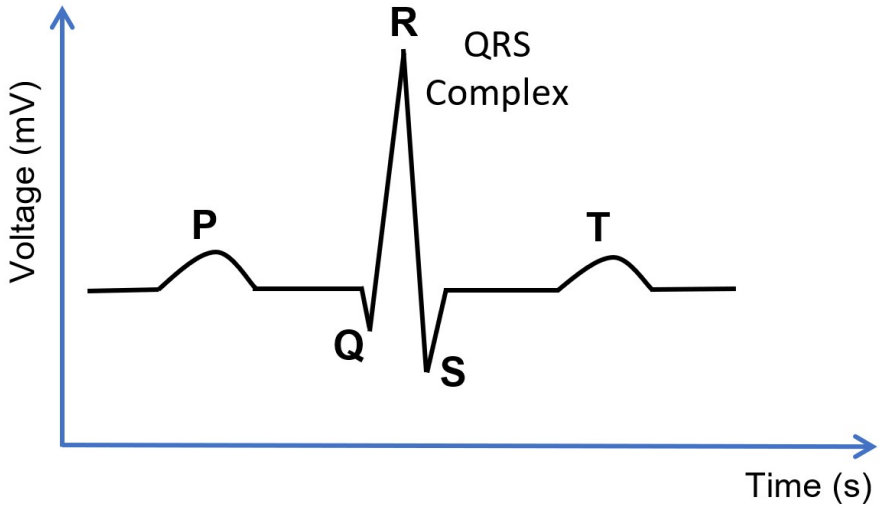

Figure 2.1: An ECG beat represented on Voltage and Time axis consisting of 'PQRST' cardiac cycle

A ECG beat represents a cardiac cycle which is a consequence of contraction and relaxation of artria and ventricles of the heart. A typical heartbeat or ECG beat of a healthy individual lasts for approximately 0.6-0.7s. A representation of ECG beat is shown in Figure 2.1. One cardiac cycle in an ECG signal can be associated with the 'PQRST' characteristic waves. The

'P' wave corresponds to atrial contraction. The 'QRS' complex corresponds to the contraction of the ventricles and the relaxation of the ventricles is seen in the form of the 'T' wave.

2.1.1. ECG Arrhythmia

Irregularity in the 'QRS' complex and its surrounding waves reflect the presence of abnormal heart activity called arrhythmias. It can be noticed by observing the morphology of the ECG beat. Features like the duration of 'QRS' complex, amplitude, distance, slope, angle and some miscellaneous features are useful for identifying arrhythmias. The 'QRS' complex is a crucial element to the identification of abnormality.

The database used in this work is the MIT-BIH Arrhythmia database [40], available Open Source in Physiobank. It is one of the most representative database for ECG arrhythmia classification. Since it has been widely used for the published research on ECG arrhythmia classification, it becomes a good standard for comparison with other works. The database consists of 30 minutes of ECG recording from 48 individual patients. Figure 2.2 shows the different types of arrhythmia classes that exists in the database. Further, Association for the Advancement of Medical Instrumentation (AAMI) recommends certain standards for a fair comparison and validation of ECG classification solutions. Among all the different types of arrhythmia that occurs, AAMI recommends the use of the arrhythmia types included in MIT-BIH database. The 15 arrhythmia types are grouped into 5 superclasses namely, Normal Beat (N), Supraventricular Ectopic Beat (SVEB), Ventricular Ectopic Beat (VEB), Fusion Beat (F) and Unknown Beat (Q). Table 2.1 shows the arrhythmia types and the hierarchy of the five superclasses.

Table 2.1: Types of ECG arrhythmia based on MIT-BIH database [40]

	AAMI Class	**MIT-BIH Arrhythmia Class**
Normal	Normal (N)	Normal Beat (N)
		Left Bundle Branch Block Beat (L)
		Right Bundle Branch Block Beat (R)
		Atrial Escape Beat (e)
		Nodal (junctional) Escape Beat (j)
Abormal	Supraventricular Ectopic Beat (SVEB)	Atrial Premature Beat (A)
		Aberrated Atrial Premature Beat (a)
		Nodal (junctional) Premature Beat (J)
		Supraventricular Premature Beat (S)
	Fusion Beat (F)	Fusion of Ventricular and Normal Beat (F)
	Ventrciular Ectopic Beat (VEB)	Premature Ventricular Contraction (V)
		Ventricular Escape Beat (E)
	Unknown Beat (Q)	Paced Beat (/)
		Fusion of Paced and Normal Beat (f)
		Unclassifiable Beat (Q)

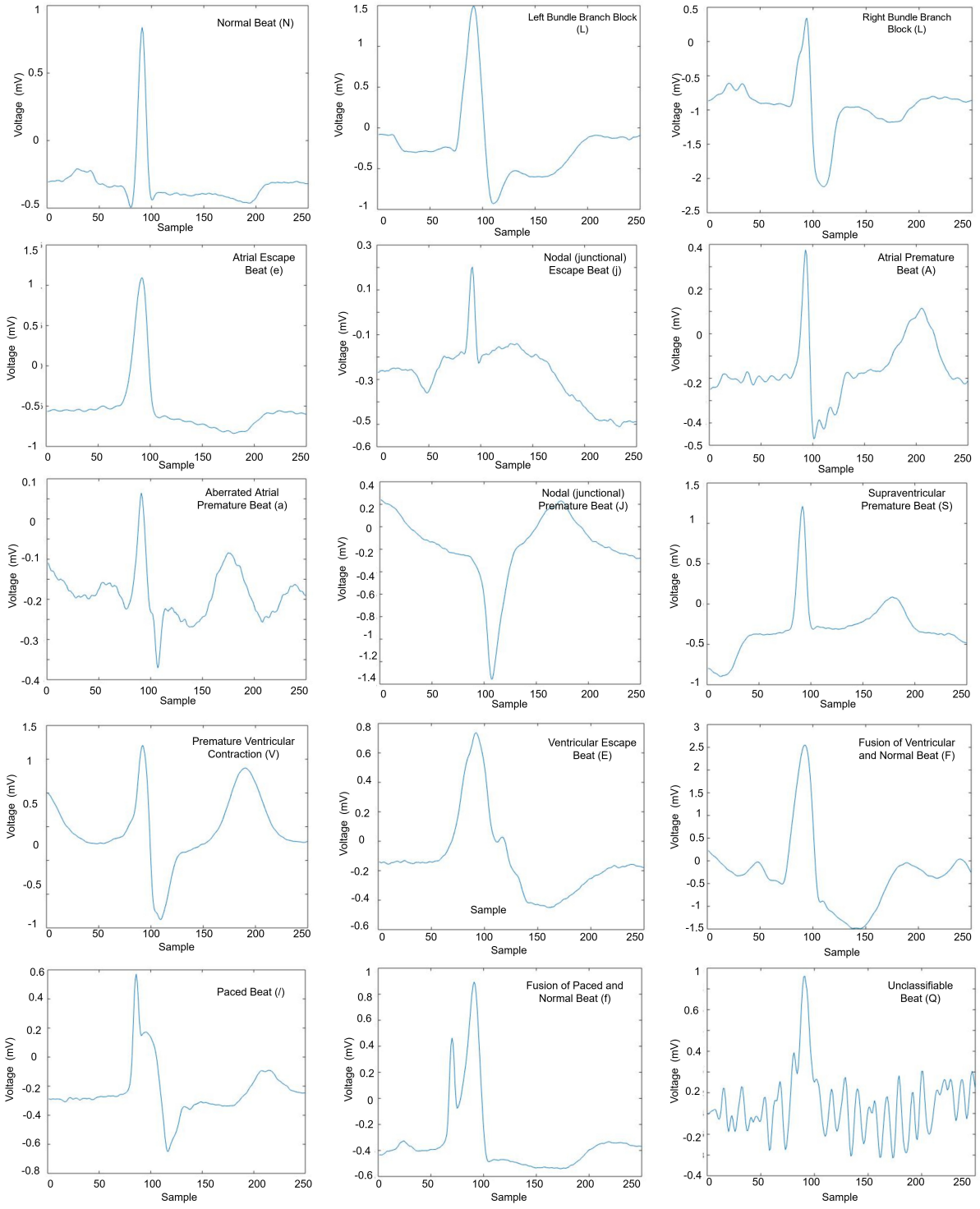

Figure 2.2: Fifteen types of ECG arrhythmia based on MIT-BIH arrhythmia database [40]

2.1.2. Classification of ECG Arrhythmia

A fully automated system for ECG arrhythmia classification can be divided into several steps - from signal acquisition to identifying the abnormality, as shown in Figure 2.3. After capturing the ECG using sensors, the signal is pre-processed. The entire recorded dataset is then divided into two parts - training set and testing set. Usually a split percentage of 80%-20% of dataset is used for training set and testing set respectively. Next, the neural network is trained and validated for the predicted output class.

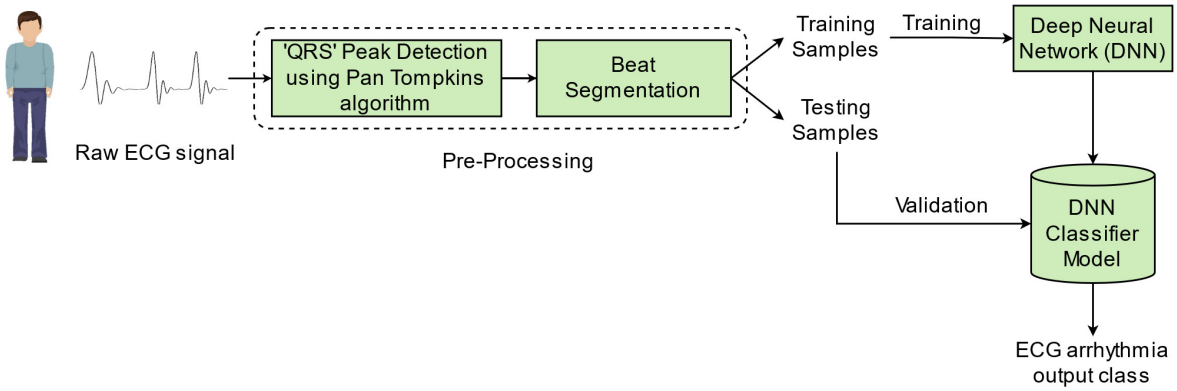

Figure 2.3: Classification of ECG arrhythmia using Deep Neural Network (DNN) starting from ECG signal acquisition to identifying the output class

2.1.3. Pre-Processing

The methods used in pre-processing stages are very essential for the classifier to have a good accuracy. The pre-processing in ECG classification is required to remove noise, identify 'QRS' peaks and segment ECG beats. It is described in more detail below:

'QRS' Peak Detection and Beat Segmentation

One of the popular methods in 'QRS' peak detection is the Pan Tompkins algorithm [43]. A flow diagram of the algorithm [43] is shown in Figure 2.4. This algorithm utilizes the amplitude, slope, and a moving window integration to identify the 'R' peaks of 'QRS' complexes in the ECG signal. In the first step, the raw ECG signal is treated with noise removal using signal smoothing, width and slope enhancement. In the next step, adaptive thresholding is used for selecting the true 'QRS' peaks and eliminating the noise peaks. The steps are described below:

- Band-pass filter: The ECG signal is passed through a band-pass filter which consists of cascaded low pass and high pass filters with 5-15 Hz passband. The LPF is used to remove high frequency noise such as powerline interference, T wave interference. The HPF removes low frequency noise such as baseline wander.
- Derivative filter: Next, the ECG signal is differentiated by a derivative filter. It suppresses the low-frequency P and T waves and highlights the slope of 'QRS' wave.
- Squaring: In order to enhance the higher amplitudes of 'QRS' complex, the ECG signal is passed through a squaring function. The squaring function helps reduce the amplitudes of T-waves that can cause false 'QRS' peak detection.
- Moving Window Integration (MWI): The MWI is used to integrate the squared signal and acquire information about the potential 'QRS' peaks. These are also known as 'fiducial marks'.

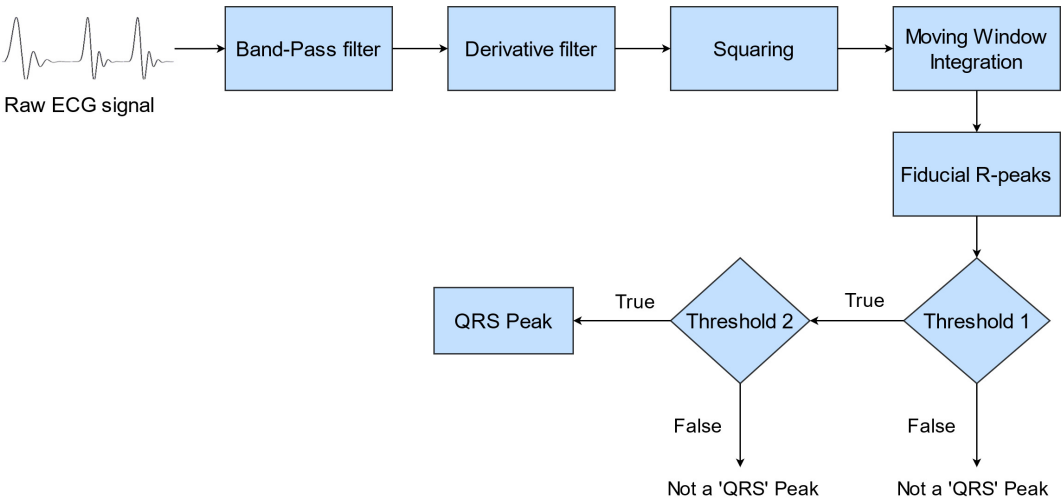

Figure 2.4: Processing steps of 'QRS' detection using Pan Tompkins algorithm

- Decision: After preparing the signal, a thresholding technique is used to detect the true 'QRS' peaks from the set of 'fiducial marks'. In the beginning, the first threshold is compared with fiducial marks' signal level. If passed, a second level of verification is carried out by comparing the filtered signal with the second threshold. Only after crossing both the stages, the right 'QRS' peak is chosen. The thresholds are automatically adjusted to improve the 'QRS' complex detection. The mathematical equations for thresholding technique as used in the original algorithm [43] is given below:

$$Threshold_1 = NoiseLevel_1 + 0.25 * (SignalLevel_1 - NoiseLevel_1)$$

initialize $SignalLevel_1$ as 0.25 of the maximum amplitude of the integrated signal and $NoiseLevel_1$ as 0.5 of the mean amplitude of the integrated signal.

The threshold is automatically updated after detecting a new peak in the integrated signal. If the new $PEAK_1$ is under the $Threshold_1$, the $PEAK_1$ is a noise peak and the $NoiseLevel_1$ is updated.

$$SignalLevel_1 = 0.125 * PEAK_1 + 0.875 * NoiseLevel_1$$

If $PEAK_1$ is above the $Threshold_1$, $PEAK_1$ is a signal peak and the $SignalLevel_1$ is updated.

$$NoiseLevel_1 = 0.125 * PEAK_1 + 0.875 * NoiseLevel_1$$

The peak corresponding to the one evaluated on the integrated signal is searched and compared with a second threshold $Threshold_2$, calculated in a similar way to the previous step for the filtered signal:

$$Threshold_2 = NoiseLevel_2 + 0.25 * (SignalLevel_2 - NoiseLevel_2)$$

$$SignalLevel_2 = 0.125 * PEAK_2 + 0.875 * NoiseLevel_2$$

$$NoiseLevel_2 = 0.125 * PEAK_2 + 0.875 * NoiseLevel_2$$

After identifying the 'R' peaks, the beats are sliced for analysis and classification. The beat is segmented such that one ECG beat contains one 'PQRST' cycle. This is achieved by taking

90 samples to the left of the 'R' peak and 159 to the right of the 'R' peak (including the 'R' peak). This forms a ECG beat of length 250 samples. ECG beat of length 250 has been widely used in several works ([20], [55], [19]). Multiple ECG beats are extracted following this approach, which form the full data set.

2.1.4. Training and Validation

Training is a process in which the neural network model learns to identify various types of ECG arrhythmia. The performance of the application largely depends on how well the network gets trained. There are two approaches for training - supervised and unsupervised training. Supervised training involves providing the network with the expected output for the given inputs. Whereas, unsupervised training requires the network to make sense of the data by itself without giving any expected output.

This work adopts a supervised training approach where the training set along with its corresponding class labels are given to the neural network. Each ECG beat consisting of 250 samples and an associated class label is given to the network. To start the training process, the weights of the network are first initialized. Then, the model processes the inputs through the network and predicts the output. The predicted output is compared with the desired output. Errors are then back propagated through the system, causing the system to adjust the weights to reduce error. This process is repeated multiple times to maximise the recognition accuracy. Once the system has been correctly trained, and no further learning is needed, the weights get finalized and are validated with the testing set.

2.2. Basics of Deep Neural Network Topologies

This section provides a basic understanding of the Deep Neural Network Topologies. DNNs such as Fully Connected Network, Long Short Term Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN) and Hybrid of TCN & LSTM are discussed below:

2.2.1. Fully Connected Network

The most basic type of feed forward neural network is the Fully Connected Network. A representation of a Fully Connected network can be seen in Figure 2.5. The network consists of an input layer, hidden layer(s) and an output layer. As the name suggests, all the neurons in a fully connected layer have connections to all the activations in the previous layer.

Input Layer — This layer is given with input vector $(x_1, ..., x_m)$. The values are processed so as to standardize them before giving it to the next layer.

Hidden Layer — At the hidden layer, the values from each input neuron is multiplied by a weight w_{ij} and a bias b_i is added to the result. The resulting weighted sum of all the values arriving at a neuron is fed into a activation function f, which gives the output of the hidden neuron h_j.

$$h_j = f(\sum_i (x_i * w_{ij} + b_i)) \tag{2.1}$$

Output Layer — The output layer is configured in terms of the number of output classes. In order to represent N output classes, N neurons are needed for the output layer. The values from each hidden neuron from the previous layer is multiplied by a weight (w_{jk}) and a bias b_j is added to the result. The resulting weighted sum of all the values arriving at a neuron is fed into a activation function f, which outputs a value y_k. The activation function can be same or

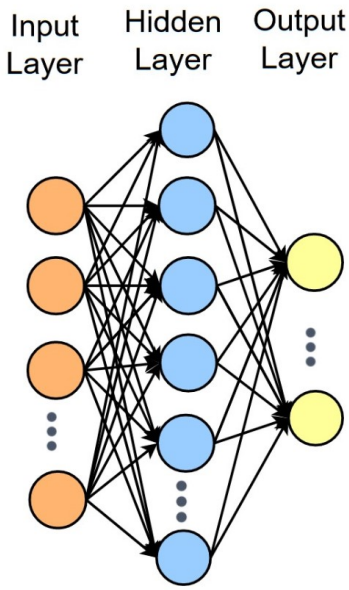

Figure 2.5: Fully Connected Network

different for each hidden layer. The y values are the outputs of the network. A softmax is used to identify the classification output.

$$y_k = f(\sum_j (h_j * w_{jk} + b_j)) \tag{2.2}$$

2.2.2. Long-Short-Term-Memory Networks (LSTM)

Long Short Term Memory (LSTM) networks are modified versions of traditional Recurrent Neural Networks (RNNs) and are replacing the latter largely. Traditional RNNs cannot provide long term dependency of deriving context from time steps deeper in the history. They often suffer from the problem of exploding and vanishing gradients. LSTMs overcome this problem using memory cells composed of gates [14], [30].

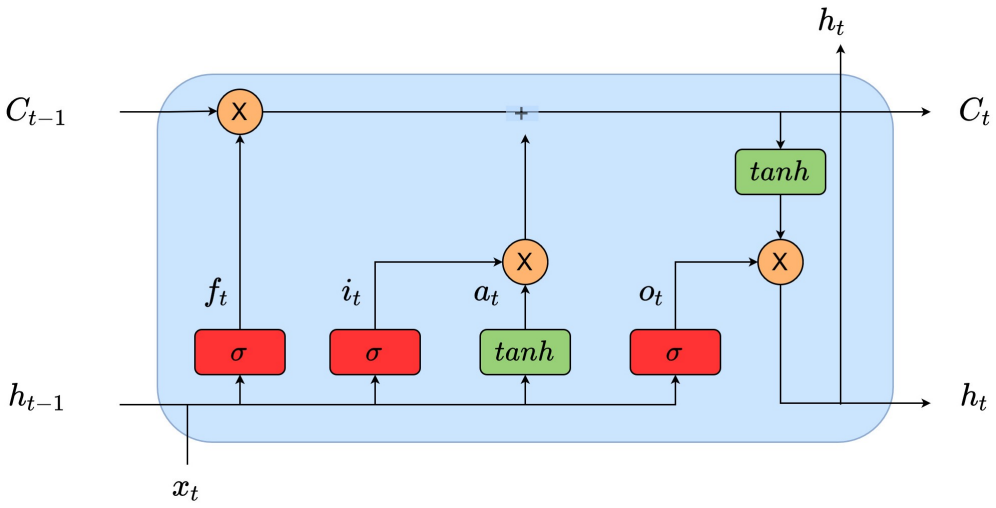

Figure 2.6: LSTM cell composed of forget gate f_t, input gate i_t, and output gate o_t

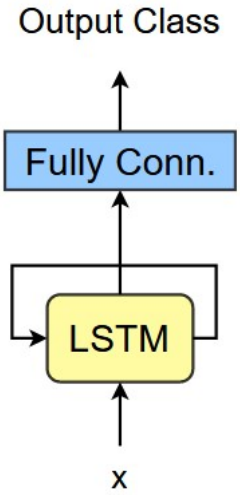

Figure 2.7: An illustration of LSTM network for classification

A LSTM memory cell is shown in Figure 2.6. It is composed of a forget gate f_t, an input gate i_t, and an output gate o_t. The forget gate f_t decides how much information is to be retained or forgotten from the memory block. The input gate i_t controls what information is to be stored in memory. The output gate o_t determines when the stored information can be used. The cell state, represented by C_t, passes information from earlier layers to faraway layers.

The computation equations for a single forward pass of LSTMs are given below,

$$f_t = \sigma(U_f * x_t + W_f * h_{t-1}) \tag{2.3}$$

$$i_t = \sigma(U_i * x_t + W_i * h_{t-1}) \tag{2.4}$$

$$a_t = tanh(U_a * x_t + W_a * h_{t-1}) \tag{2.5}$$

$$o_t = \sigma(U_o * x_t + W_o * h_{t-1}) \tag{2.6}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot a_t \tag{2.7}$$

$$h_t = o_t \odot tanh(c_t) \tag{2.8}$$

where U_f, U_i, U_g, U_o denote the weight matrices for the input vector, and W_f, W_i, W_g, W_o represent the weight matrices of the hidden state of the time step.

2.2.3. Bidirectional Long-Short-Term-Memory Network (BLSTM)

LSTM structures process previous information rather than any subsequent information and are unidirectional in general [61]. To allow information flow from later time steps, Bidirectional LSTMs (BLSTM) are used. Figure 2.8 shows a bidirectional LSTM model. It contains two LSTMs, processing information in opposite directions, forward and backward. To generate the classification output at the last time step, the output from the two LSTMs connect at the Fully Connected layer.

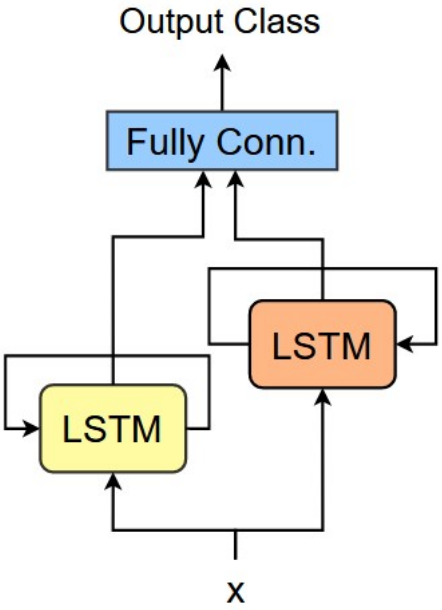

Figure 2.8: An illustration of BLSTM network for classification

2.2.4. Temporal Convolutional Network (TCN)

Temporal Convolutional Network (TCN) are inspired by convolutional architectures for sequential data. TCNs use 1D fully-convolutional network architecture. The architecture can take a sequence of any length and map it to an output sequence of the same length. The architecture of TCNs used in this work is based on the architecture proposed by [11] in 2018.

The basic building block of a TCN is a Residual Block, as shown in Figure 2.9. TCNs consists of an input layer, an output layer, and multiple hidden layers of residual blocks in-between. It is followed by fully connected layer to perform the classification. Residual block consist of dilated causal convolution layers in addition to weight normalization, non-linear ReLU and dropout layers. The number of residual blocks n to be stacked depends upon the receptive field size (RF) and kernel size k_size of the application [11], which follows the relationship:

$$RF = 1 + 2 * (k_size - 1) * 2^{(n-1)} \tag{2.9}$$

Dilated Causal Convolutional Layer

The dilated causal convolutional layer is the core of a residual block and does most of the computational operation. A convolution layer defines a filter to examine a subset of the signal. The filter is subsequently translated through the entire signal which helps to parameterize specific features within the entire signal. The filter is also known as a kernel. The output of the convolution operation is referred to as a feature map. The filter dimension depends upon the application. For example, in case of a signal, 1D filters are used and whereas 2D filters are needed for images. Causal convolutions at time t are result of convolutions with elements from time t and earlier in the previous layer. Causal convolutions make TCNs suitable for sequential applications such as ECG. Using a dilated convolution helps to achieve a larger receptive field with fewer layers. Dilated convolutions are a type of convolution that expand the kernel by inserting holes between the kernel elements.The dilation parameter indicates how much the kernel is widened by inserting spaces between kernel elements. For a 1-D input sequence x and a filter f (size: k_size), the dilated convolution operation F on element

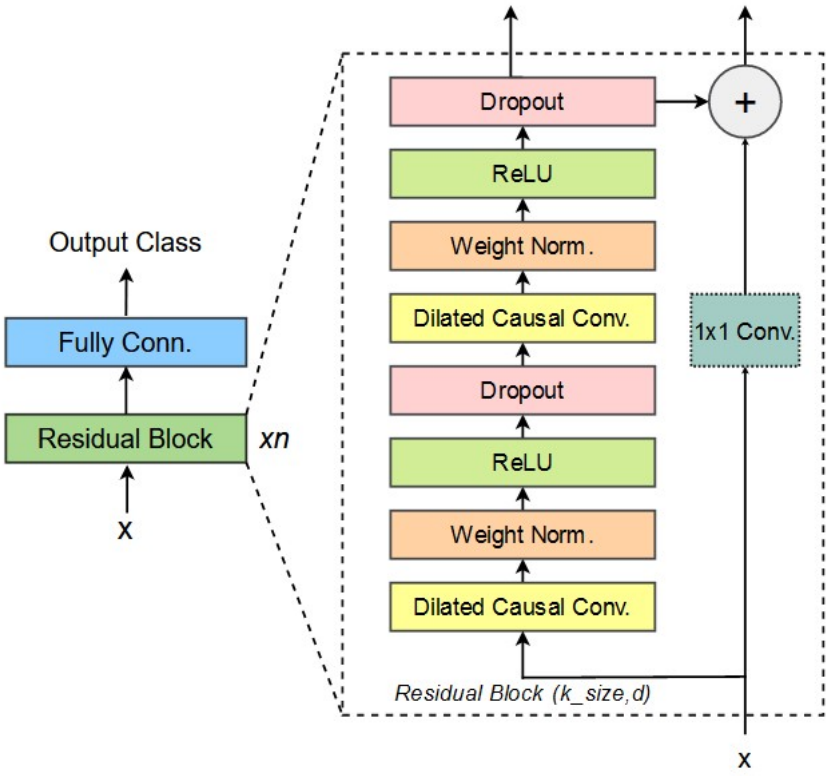

Figure 2.9: An illustration of TCN network for classification

s of the input sequence can be defined as:

$$F(s) = x *_d f(s) = \sum_{i=0}^{k_size-1} f(i).x_{s-d.i} \tag{2.10}$$

In case the width (number of channels) of the inputs and the width (number of filters) of the second dilated causal convolution layers differs, a 1D convolution is applied to the input and added to the convolution outputs to match the widths.

Weight Normalization Layer
TCNs use weight normalization layer after dilated causal convolution. Weight normalization is used to decouple the length of the weight vector from the direction.

Dropout Layer
In a dropout layer, some number of layer outputs are randomly ignored during training. This changes the layer configuration and makes the layer look-like a layer with a different number of nodes and connectivity to the prior layer. It approximates training a large number of neural networks with different architectures in parallel, and reduces overfitting.

2.2.5. Hybrid Neural Networks

A novel architecture of combining TCN and LSTM was also explored in this work. The structure is shown in Figure 2.10. The first layer is a TCN residual block followed by LSTM layer. Combining the two architectures is done to analyze how effective feature extraction and classification can be done by the combination of the two types.

Performance comparison of the DNN topologies is discussed in detail in Section 4.4.

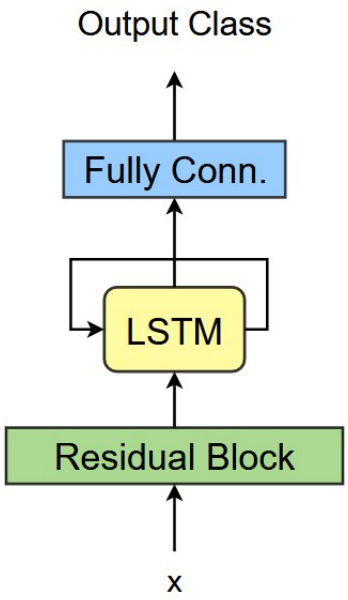

Figure 2.10: An illustration of hybrid architecture combining TCN and LSTM

2.3. Dimensionality Reduction Techniques

Dimensionality reduction refers to the process of reducing the number of features of a dataset. Reduction in number of features is done ensuring variance of the dataset is maintained as much as possible. Few of the most widely used dimensionality reduction techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Discrete Wavelet Transform (DWT) are described below:

2.3.1. Principal Component Analysis (PCA)

PCA is a statistical technique of unsupervised dimensionality reduction which captures variance in data by representing variables in directions with largest variance known as 'principal components'. The first principal component is along a direction which maximizes the variance of the projected data the most, followed by the second principal component with second highest variance etc. [5]. The algorithm has a time complexity of $O(N^2p)$, where N is the total number of beats and p is the total number of features in each beat. The steps to compute principal components of a dataset are as follows [1]:

1. Standardization of data: All the variables contribute equally to the PCA computation process, therefore it is important to standardize them to the same scale [1]. Standardization is done by removing the mean of the data samples from the value and dividing by its standard deviation. The general expression for standardization can be seen in Equation 2.11.

$$y = \frac{x - mean}{std.dev.} \tag{2.11}$$

2. Covariance matrix computation: In order to analyze if there is any relationship between the variables, covariance matrix is computed. In case the variables are highly correlated, the covariance value is positive. This implies presence of redundant information and vice-versa [5].

3. Principal Component computation: Next, the eigen values of the covariance matrix are

computed. The eigen values are sorted in decreasing order with each eigen value having reduced variance contribution to the data. The transformation takes place such that $Y = W^T X$, where W is the set of eigen values, Y is the set of principal components and X is the input matrix. The maximum amount of information is contained in the first few principal components. While doing dimensionality reduction, only the first few components are retained and the others are discarded [1].

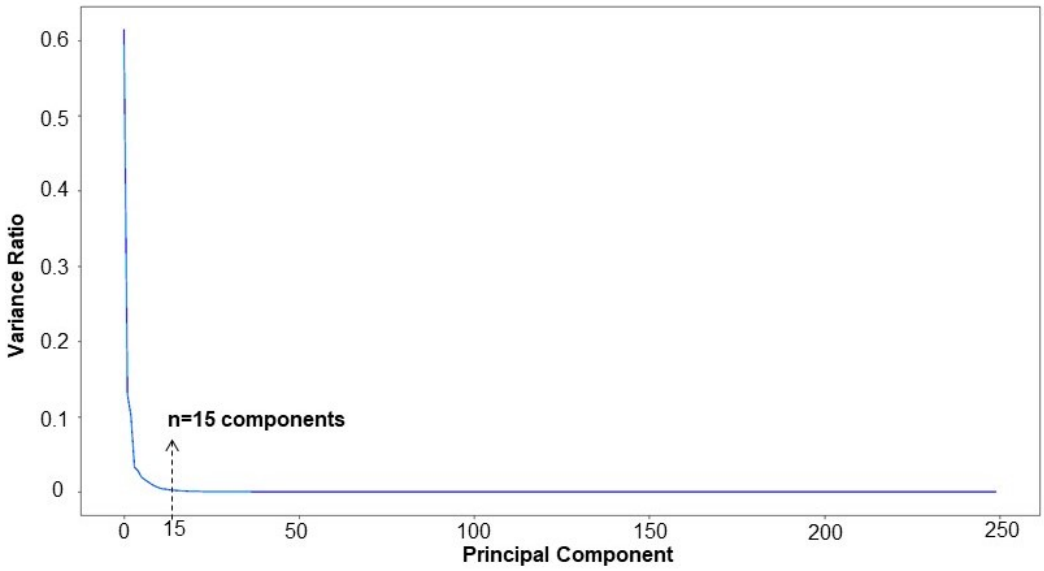

Figure 2.11: Decreasing variance ratio of principal components computed for dimensionality reduction of ECG beat. After n=15, the variance contribution of each principal component becomes close to zero

Figure 2.11 shows the variance ratio of each of the principal components for a single ECG beat of length 250. It is clear that the significance of the principal components reduces after n=15. Therefore, 15 significant principal components were chosen to represent a ECG beat.

2.3.2. Linear Discriminant Analysis (LDA)

LDA is closely related to PCA as they both look for linear combinations of variables to represent the data. But LDA is a supervised technique where it not only minimizes variance within each class but also maximizes variance between classes [4]. It does the variable separation by computing the directions, linear discriminants, representing the axis maximizing the separation across classes. The algorithm has a time complexity of $O(N^2p)$, where N is the total number of beats and p is the total number of features in each beat. The steps to compute linear discriminants of a dataset are as follows [6]:

1. Between-class covariance matrix: Computation of the between class covariance matrix is done by calculating the distance between the class mean and global mean. This gives an estimate of separability between different classes [6].

   The expression for finding between class variance is given in Equation. 2.12.

$$S_{between} = \sum_{i=1}^{num_classes} N_i(\bar{x_i} - \bar{x})(\bar{x_i} - \bar{x})^T \tag{2.12}$$

2. Within-class covariance matrix: In a similar way, within class covariance matrix is computed by calculating the distance between the class means and class samples and adding them [6].

   The expression for finding the within class variance can be seen in Equation. 2.13.

$$S_{within} = \sum_{j=1}^{N_i} \sum_{i=1}^{num_classes} N_i(x_{i,j}^{-} - \bar{x_i})(x_{i,j}^{-} - \bar{x_i})^T \tag{2.13}$$

3. The values are projected to a lower dimensional space so as to maximise between-class variance and minimize within-class variance. LDA computation returns 4 linear discriminants representing an ECG beat of length 250.

2.3.3. Discrete Wavelet Transform (DWT)

DWT is commonly used to examine biomedical signals as it represents both time and frequency domain information of the signal in a multi-resolution manner [3]. The ECG signal is convolved with a discrete wavelet and down sampled by 2 each time. The operation is represented using a filter-bank as shown in Figure 2.12. The input x is passed through a high pass filter h and a low pass filter g. It is then down sampled by 2 giving a set of 'detailed' (D) and 'approximation' (A) coefficients at each level. This process is repeated multiple times depending upon the required resolution of samples. The expression for the computation is given in Equation. 2.14 and 2.15. The algorithm has a time complexity of O(N).

$$y_{low}[n] = \sum_{k=-\infty}^{+\infty} x[k] * g[2n-k] \tag{2.14}$$

$$y_{high}[n] = \sum_{k=-\infty}^{+\infty} x[k] * h[2n-k] \tag{2.15}$$

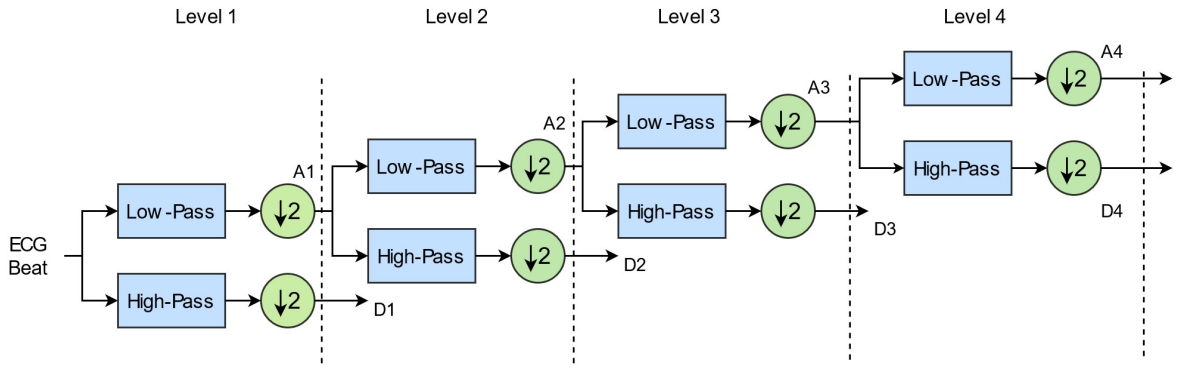

Figure 2.12: A four level ECG beat decomposition into Approximation (A) and Detailed (D) coefficients for performing DWT

Daubechies2 (db2) filter with a level 4 decomposition is used in the thesis. Db2 is widely applied for ECG signal decomposition ([55], [27], [60]). It shows better performance for ECG than other similar discrete wavelets [27] (symlet, coiflet etc.). The db2 low-pass and high-pass filter coefficients are given below and shown in Figure 2.13. :
Low Pass = [-0.1294, 0.2241, 0.8365, 0.4829]
High Pass = [-0.4829, 0.8365, -0.2241, -0.1294].

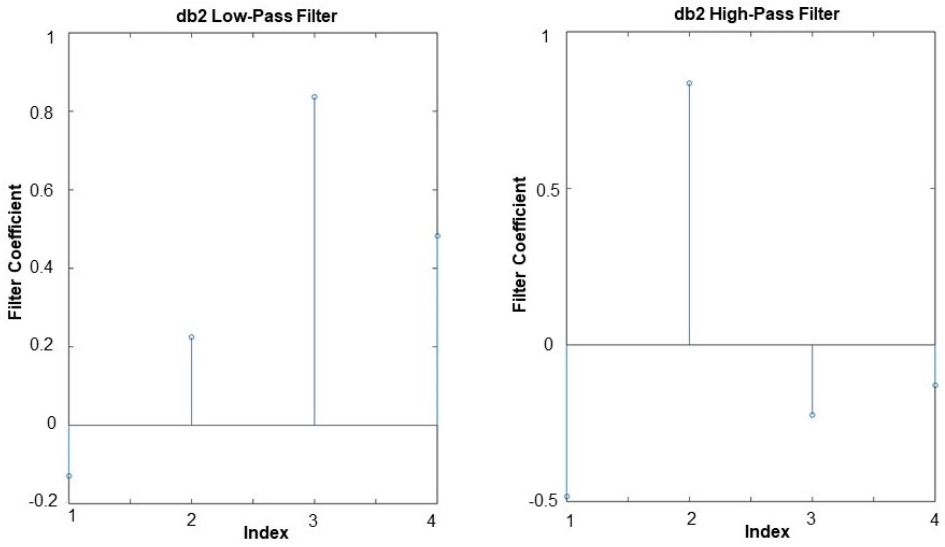

Figure 2.13: Low Pass and High Pass decomposition filter for db2 wavelet

After computing the approximate and detail coefficients at each level, statistical features such as minimum, maximum, mean, variance and standard deviation of the coefficients can be used to represent ECG data. This is a similar approach used in [27] and results in a feature vector length 25 for an ECG beat.

2.4. Computation in-memory using RRAM devices

Computation in-memory (CIM) is a highly effective method to mitigate the 'memory wall' problem of conventional von Neumann architectures. A representation of the von Neumann and CIM architectures is shown in Figure 2.14.

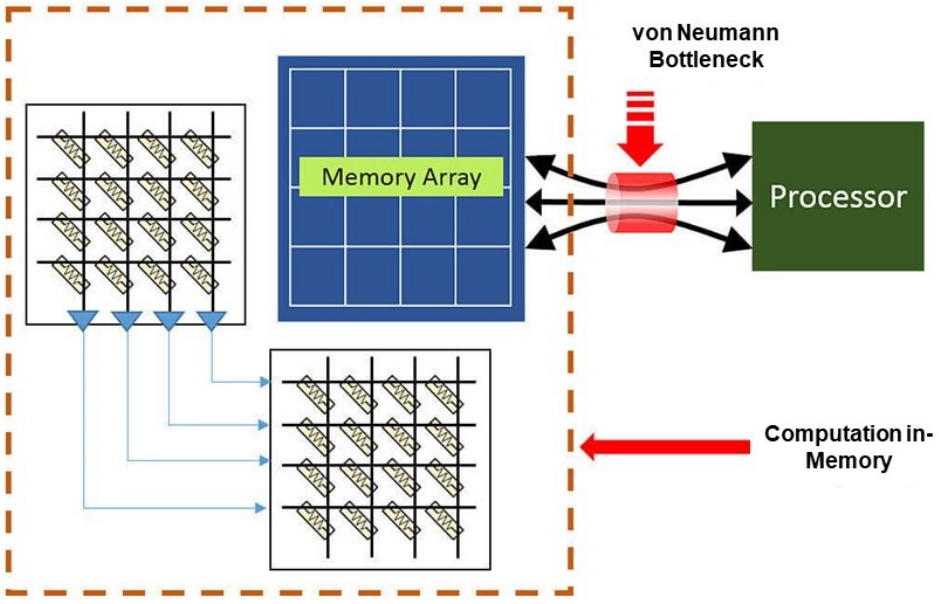

Figure 2.14: von Neumann bottleneck in conventional computing overcome by Computation In-Memory (CIM) architectures [17]

While von Neumann architectures require data movement between memory and processor, CIM reduces the data movement by allowing data storage and computation at the same memory location. CIM has shown promising potential for wide range of complex and cognitive tasks like visual/audio recognition, self-driving etc. [25].

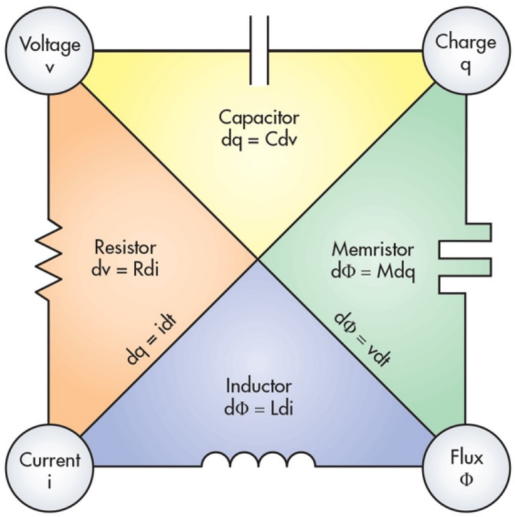

Figure 2.15: Four fundamental two-terminal circuit elements relating Voltage (V), Charge (q), Current (i) and Flux (ϕ) [2]

Memristors are the building blocks of CIM architectures. As the name suggests, it is a non-linear resistor element with 'memory'. It was first proposed by Leon Chua in 1971 [15], and first demonstrated by Hewlett-Packard (HP) Labs in 2008 [52]. The hypothesis came from the fact that there is a missing two-terminal circuit element, characterized by a relationship between the charge $q(t)$ and the magnetic flux-linkage $\phi(t)$. It was introduced as the fourth basic circuit element as shown in Figure 2.15. Since then, research at several levels, including evaluating different metal oxides, their physical properties, development of circuits and architectures etc. have captured great attention of the research community.

The memristive devices are non-volatile memories (NVMs). There are several types of memristive devices, such as resistive random access memory (RRAM), phase change memory (PCM), spin-transfer-torque magnetic RAM (STT-MRAM) etc. RRAMs is considered in the thesis.

RRAMs store the network weights in the form of memristive conductances and perform computation within the memory. They are being explored to make the neural network applications more efficient in terms of speed, computational complexity, area, energy consumption etc. By reducing the need for data movement between the memory and computation unit, improvements of upto 5.5x in energy and 7.5x in computational density can be obtained [51]. Individual memristors performing computation of logic lead to massive parallelism in the system. Thus, compared to the existing computer architectures, CIM provides a more energy efficient solution for neural network applications.

2.4.1. Memristor Operation

The memristor relates flux and electrical charge. It behaves similar to a resistor but with a non-linear behaviour known as memristance. It follows the relation:

$$M = \frac{d\phi(t)}{dq(t)} = \frac{d\phi(t)/dt}{dq(t)/dt} = \frac{V(t)}{I(t)} \tag{2.16}$$

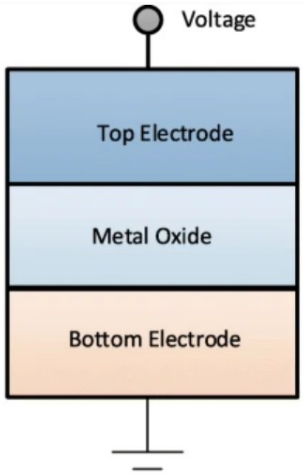

Figure 2.16: Schematic of a metal oxide RRAM based on metal-insulator-metal structure [25]

A RRAM device consists of a resistive switching memory cell having a metal-insulator-metal structure. Schematic of a Metal oxide RAM is shown in Figure 2.16. It consists of top and bottom metal electrodes. The bulk of the device is made up of metal oxides such as TiO2 [25]. The oxide layer consist of oxygen vacancies which act as conductive filament between the 2 electrodes. When an external voltage is applied across the RRAM cell, the device switches from High Resistance State (HRS) or off-state with resistance R_{off} to a low resistance state (LRS) or on-state with resistance R_{on}.

The voltage to switch the RRAM cell from LRS to HRS happens on application of a RESET voltage or V_{RESET} and the process is referred to as the 'RESET' process. Likewise, the HRS of the RRAM can be changed to LRS on the application of the voltage referred to as SET voltage V_{SET} and the process is referred to as the 'SET' process. The SET and RESET process of the RRAM is shown in Figure 2.17. Since both LRS and HRS retain their respective values even after the removal of applied voltage, RRAM is a non-volatile memory.

When a RRAM device is manufactured, it is initially in HRS. The resistive switching in the RRAM happens only on application of external voltage which leads to breakdown of the metal-insulator-metal structure of the cell. It forms a conductive path in the switching layer and the cell moves into LRS [25], [39]. The process is referred to as 'electroforming'. The voltage at which this occurs is known as the forming voltage, V_f [25].

2.4.2. RRAM devices as crossbar array

The CIM architecture uses the memristive devices arranged in a crossbar array structure to perform computations for different applications. The fundamental operation of all the neural network computations is a Multiply and Accumulate (MAC) operation. It also known as a Vector Matrix Multiplication (VMM). The resistive state or the conductance level of the RRAM is modified to represent network weights, by applying voltages and currents. However, the bit-precision of memristor cells are usually less than that required by the neural network weights. Therefore, bit-slicing technique is utilized, where the bits are divided into smaller chunks. The bit-sliced network weights are mapped to the memristor conductances. The ISAAC architecture [51] considers a 16-bit resolution for input voltages and network weights. The 16-bit weights are mapped onto group of 8 RRAMs, with each RRAM programmed with a 2 bit-slice as shown in Figure 2.18. Each 16-bit input is converted to analog voltage and given to the crossbar with 1-bit at a time.

Figure 2.18 shows the mapping of a VMM operation onto a memristor crossbar array struc-

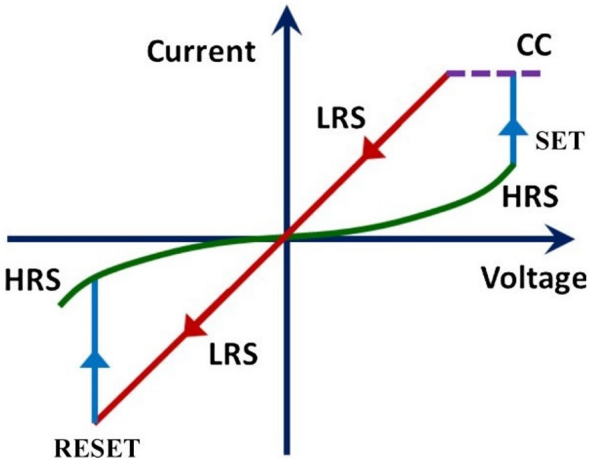

Figure 2.17: Characteristic IV curve of a RRAM device consisting of SET and RESET processes [45]

ture, with each memory element in the cross section of vertical and horizontal rails. The multiplication operation of a VMM is implemented by leveraging Ohm's law on the crossbar array. In a single clock cycle several multiplications are computed in parallel, where at each cross point, the current is the product of the input voltage and the memristive conductance, $I = V * G$. Similarly, the addition operation of the VMM is computed on a crossbar using Kirchoff's current law. At the end of each column, the total current is the summation of all the currents at each crosspoint, $I_n = \sum_{k=1}^{m} i_k$. Currents at the end of each column is converted to digital, shifted and added to get the final output.

RRAM devices suffer from reliability issues. It becomes a major concern when integrated in large scale for CIM applications. These issues arise from not only device fabrication process but also during the computation process. Process variations causes RRAM devices to have varying conductance range, device programmability, retention and endurance. This leads to faults and can reduce the yield. During computations and device programming, repeatedly rewriting the devices could deteriorate the system performance. The bit error rate of a RRAM device is directly linked to switching time which is related to the device characteristics. In order to get a reduced effect of process variations, a higher margin of voltage or longer duration of pulses may be applied to the RRAM devices. Other techniques to avoid process variations include using error correcting codes or parity bits to correct and detect errors. These techniques however can lead to higher area and power overhead.

Long time use of RRAMs and repeatedly applying voltages to cells can lead to read disturbance errors. It causes unexpected weight drifts from the original trained values. Usually a read-verify scheme is used for programming RRAM devices. It helps in lowering the bit error rates. Additional retraining of weights using feedback loops can improve weight stability. These loops depend upon quantifying the amount of drift in the previous cycle and adjusting the applied voltage such that it takes into consideration the drift value.

To address the reliability concerns of RRAM devices in applications, it is widely accepted to assume that these variations follow a Gaussian distribution [28], [58]. [28] used $G_p = G_d + \nabla G$ where G_p is the final programmed conductance, G_d is the desired conductance value, and $\nabla G \sim N(\mu, \sigma)$ is a random variable that follows a Gaussian distribution of mean value μ and standard deviation σ.

The precision level achievable with the crossbar computation is defined by the number of possible conductance levels in the RRAM. However, the difficulty arises in accurate tun-

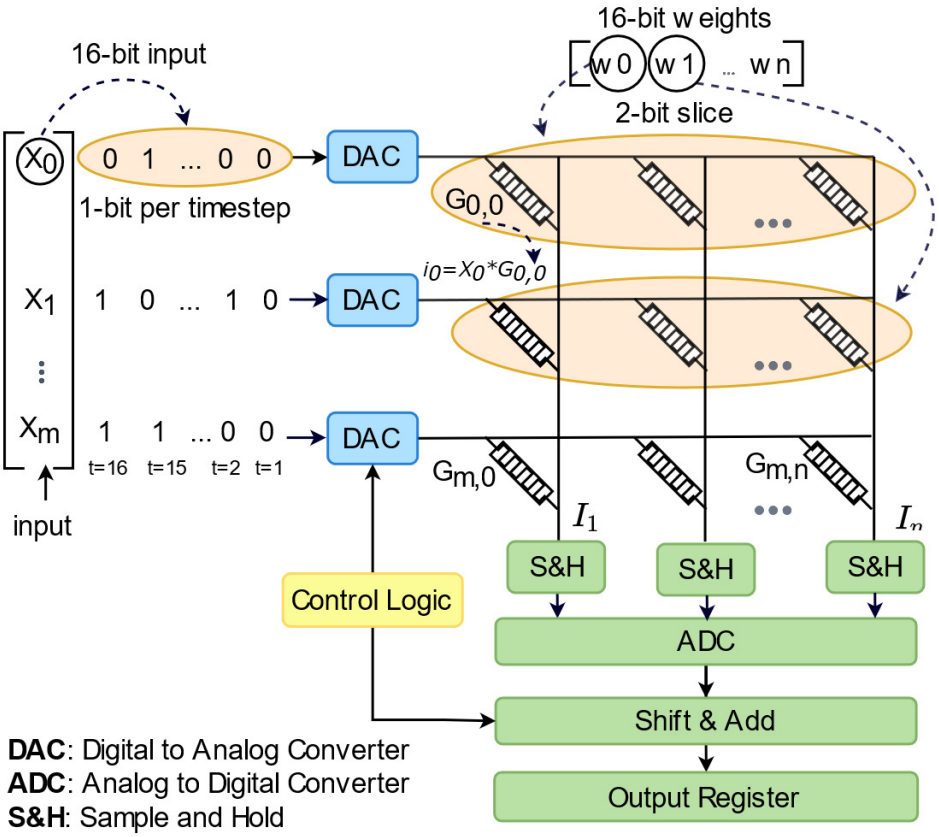

Figure 2.18: RRAM based DNN framework consisting of RRAM crossbar array and periphery devices, similar to ISAAC [51]

ing/modelling of the conductance values because of several non-idealities present such as parasitic sneak path currents (current flowing through unselected cells), asymmetric switching of conductance values upon application of current and voltage pulses etc. Several techniques have been proposed in order to reduce the sneak path currents. One such method is to use two terminal selectors such as a transistor or diodes. When a RRAM is used in series with a transistor or a diode, the crossbar array is called active 1T1R or passive 1S1R crossbar array, respectively [46]. Due to smaller size of the diodes, they are preferred over transistors for area limited applications. Sneak path currents also limit the size of crossbar array. Using selectors to reduce sneak path current also has few drawbacks. Diodes themselves have non-linear characteristics and may affect the memristor in its operating region.

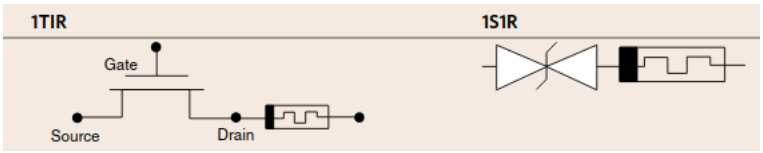

Figure 2.19: 1T1R and 1S1R cell [46]

Another challenge while tuning the RRAM conductance level is that the same bias applied to different word lines is seen differently, due to voltage drop along the word and bit lines of the array structure. This can be avoided by limiting the size of crossbar arrays. Linear and symmetric switching of conductance values is crucial while programming RRAM crossbar

arrays for DNNs, i.e. the conductance should increase and decrease in the same amount on application of an electric pulse in opposite polarities. This is usually handled by using 2 devices with opposite polarity to represent single weight. Another approach to address this is to use range shifting, wherein half the weights is used to represent positive weights and other half is used for negative weights [31]. While the first method provides a higher precision, second allows for a better area efficiency [31]. The deep neural networks can easily be implemented on a RRAM crossbar array using discrete conductance levels to represent synaptic weights. In order to program the conductance level, varying voltage amplitude [23], [54] or pulse widths [49] are usually used to program the device. The precision of the crossbar array also depends on the number of conductance level. In order to achieve a higher precision > 8 bits, usually the weights are spilt into multiple bits and mapped to separate RRAM devices.

2.5. Design Metrics

For any kind of classification task, a confusion matrix is often used to describe the performance of the classifier. Figure 2.20 shows representation of a confusion matrix for a binary classification. A binary classification here means, the output of the neural network model can be classified into two classes, Normal and Abnormal. A similar approach is applicable for a multi-class classification. In such a case, the multi-class data is considered to be binarized by transforming the data under a one-vs-all scheme.

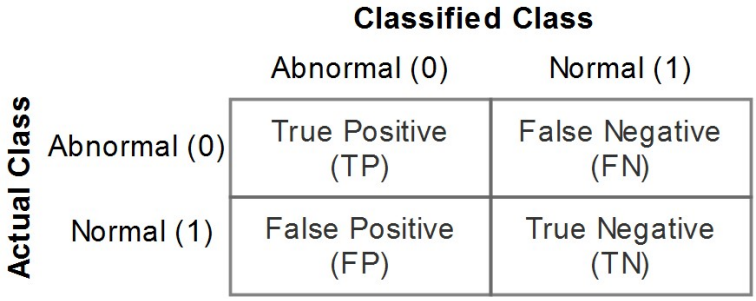

Figure 2.20: Confusion matrix for a binary classification task

- True Positives (TP): These are cases in which the beat is Abnormal and the model classifies as Abnormal.
- True Negatives (TN): These are cases in which the beat is Normal and the model classifies as Normal.
- False Positives (FP): These are cases in which the beat is Normal but the model classifies as Abnormal. It is also known as a ‘false alarm’.
- False Negatives (FN): These are cases in which the beat is Abnormal but the model classifies as Normal. It is also known as a ‘missed detection’.

The design metrics considered are at two levels - software and hardware metrics. Software metrics are accuracy, F1-score, and runtime; Hardware metrics are energy and area.

- **Software Metrics**

  - **Accuracy**: It is the ratio of correct classifications to the total number of classifications.
$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$
  - **F1 Score**: It is a very essential evaluation metric for medical applications. False Negative which is a missed detection and False Positive which is a false alarm are crucial in these classification tasks. Also, in case of imbalanced datasets, e.g. when abnormal categories occur much less frequently compared to normal category, F1 score is a better evaluation metric to validate the model's performance.
$$F1 - Score = \frac{TP}{TP + 0.5 * (FP + FN)}$$
  - **Runtime**: Runtime measures the execution time taken for inference of one ECG beat. This is a system dependent metric but gives a rough estimate of speed comparison between various neural networks.

- **Hardware Metrics**
  - **Energy**: Advances in technology have brought us tiny wearable ECG devices with efficient algorithms for anomaly detection. However, energy consumption is still a critical concern for practical usage. An ECG recording device needs to process activity of the heart for long duration of time. Besides, analysing these signals using neural networks requires a significant amount of energy consumption. While more importance is usually given to the correct functionality of the device, there is a lack of focus on energy consumption. The energy consumed by the DNN accelerator, which include the RRAM crossbar and the associated periphery devices, to process and classify the input ECG beat(s) is the most important hardware design metric.
  - **Area**: Apart from wearable devices being energy efficient, they should consume less area in order to be portable and handy. The total area occupied by the DNN accelerator composed of the RRAM crossbar and periphery devices is the second hardware design metric.

F1 score and energy consumption are the two most important evaluation metrics to validate the performance of a ECG classifier.

3

Related Works

This chapter presents a detailed review of the related works for the ECG signal arrhythmia classification. Section 3.1 presents the traditional arrhythmia classification techniques followed by Sections 3.2-3.6 discussing deep neural network techniques. A summary of the works is presented in Section 3.7.

3.1. Traditional Classification Techniques for ECG Arrhythmia

Cardiologists need to manually examine long durations of ECG recordings to identify the abnormal beats. This task is time-consuming, tedious, subjective, and costly. Sometimes the diagnosis is performed many days after the recording due to heavy workload on the doctors which can make it difficult for the patients to know their heart problems promptly, increasing the risk of delay in the medical treatment.

With the emergence of automatic computer-based ECG classification solutions, the above problems with manual investigation is solved. Machine learning algorithms such as Support Vector Machine (SVM) [48], K-nearest neighbours, fuzzy classifier [21], decision trees [50], [34] have been explored for classifying ECG arrhythmia. Zhou *et. al.* [34] use entropies of the ECG beats as feature vector for classification using random forest which operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes. They obtain an accuracy of 94.61% for classification of ECG arrhythmia generalized to new patients. In a similar approach Sadiq *et. al.* [50] derive segments of length from 'PQRST' ECG beats such as 'QRS' interval, 'QRS' amplitude, 'P' wave interval, 'T' wave interval, 'PR' interval and 'ST' interval and form the feature vector for classification using decision tress. They obtain an accuracy of 94%. However, they classify ECG arrhythmia only into five arrhythmia classes. Ray *et. al.* [48] uses SVM as classifier which aims at constructing a hyperplane separating multiple arrhythmia classes from each other. They use 'RR' interval and particle swarm optimization technique for tuning the learning parameters of the SVM classifier and obtain a high accuracy of 98.83% for classification into sixteen types of arrhythmia beats and heart rhythms. In a similar approach, Acharya *et. al.* [21] derive 'RR' interval and amplitude of 'R' peaks as features for classification using fuzzy classifier.

As can be seen, these traditional pattern recognition techniques require manual feature extraction of ECG signals to be done before processing it through the classifier model. Besides, if the feature extraction is not precise, it will eventually have a negative impact on the classification results. Therefore, in recent times more advanced algorithms using deep neural networks (DNN) are used for ECG arrhythmia classification. The remaining part of the chapter reviews the DNN implementations including Convolutional Neural Network (CNN), Long Short Term Memory Network (LSTM), Bidirectional LSTM (BLSTM), Hybrid of CNN & LSTM, Tem-

poral Convolutional Network (TCN) and Spiking Neural Network (SNN) for ECG arrhythmia classification and analyses the contributions and shortcomings in these solutions.

3.2. Convolutional Neural Network (CNN)

Rajpurkar *et. al.* [42] use a convolutional neural network (CNN) for ECG signal classification by using a non-invasive and continuous monitoring device called the Zio Patch, annotated by cardiologist experts. The network takes as input a time-series of raw ECG signal, and outputs a sequence of label predictions. To deal with variable length ECG signal, their model accepts ECG signals having length multiples of 256. If the length varies, the signal is truncated to the nearest multiple. Therefore, a unique method of classifying ECG signal input with variable lengths is done. However, F1 score of 77.6% is obtained which is less compared to other implementations. Besides, the architecture uses a large neural network which will take higher processing time and also cause higher energy consumption in hardware.

A similar technique of capturing ECG signal in real time using a wearable ECG monitoring device and classifying using CNN is described by Lin *et. al.* [35]. The captured signal is decoded, pre-processed and sent to the classifier. In a unique technique, the pre-processing stage converts the recorded ECG signals to 2D images. They also propose an APP and cloud based display plus storage of real time ECG signal captured from electrodes. The device level classification is done with the APP. To consume less power on the device, a binary classification is done while subcategory classification is done on the cloud. Without a 'QRS' peak detection, they slice ECG segments of length 1200 and classify into four global arrhythmia classes - N, SVEB, VEB and F. They obtain a less accuracy of 95.73% for MIT-BIH database and 94.96% for clinical trials. Besides, if more than one arrhythmia type is present in a single long ECG segment, network will not be able to detect them.

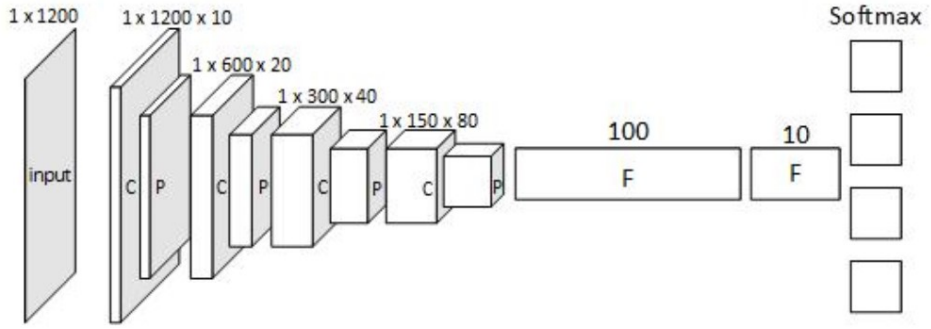

Figure 3.1: Network architecture used in [35]

Wang *et. al.* [56] use a Multilayer Perceptron Network, also known as MLP followed by a CNN for arrhythmia diagnosis. Since the MIT-BIH database is a highly unbalanced dataset, with majority of the signals being Normal and only few falling under the abnormal classes, they propose a two stage neural network (TSNN) to reduce the computational complexity and power consumption. The network structure is shown in Figure 3.2. The first-stage is a MLP network, which directly takes an ECG beat as input, to classify beats into only normal and abnormal categories. Then the abnormal heartbeats are filtered out and fed into the second stage made of CNN for further classification. It obtains an accuracy of 98.4%. Since their model is to be used for wearable devices, they propose a diagnosis-based adaptive compression scheme. Considering the trade off between the lossy (high compression ratio with loss of information) and lossless compression (more information with less compression ratio) tech-

niques, they suggest that in order to transmit the signal to the end device, the normal signal should be compressed with lossy technique and abnormal beats with a lossless technique. It uses complementary metal oxide semiconductor (CMOS) implementation for energy estimation. However, the work does not have a finer level classification of ECG beats. Besides, the large network architecture consumes high energy in hardware implementation.

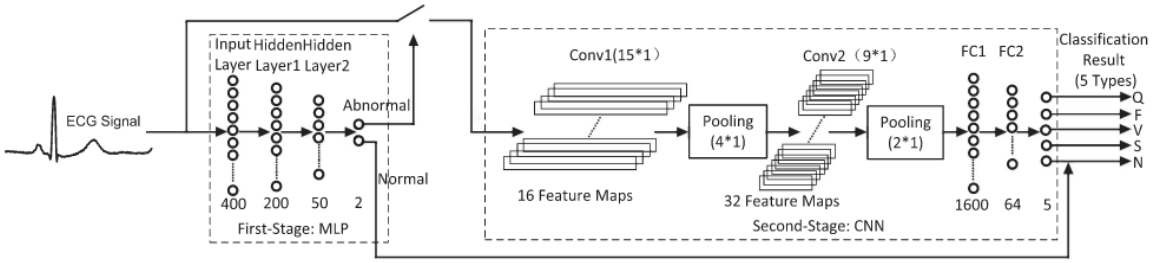

Figure 3.2: Two Stage Neural Network as proposed in [56]

A memristor based ECG signal classification is demonstrated by Hassan *et. al.* [28] for five different arrhythmia types. They use a simple two layer MLP network to do the classification. They use modified offline training using gradient descent momentum with adaptive learning rate. The ECG signal is used directly without any 'QRS' peak detection. After training, the non-idealities of the memristor device is modelled using a Gaussian variable with mean μ and standard deviation σ. The conductance matrix is modified as, $G_p = G_d + \nabla G$ where G_p is the final programmed conductance, G_d is the desired conductance value, and $\nabla G \sim N(0, \sigma)$ is a random variable that follows a Gaussian distribution. Two memristor devices are used to model each conductance weight. They have approximated the activation function $tanh$ as the summation amplifier used at the end of each column of the crossbar array. However, this work uses only a subset of arrhythmia classes: N, L, R, A, V for classification and obtain an accuracy of 96.17%. Higher accuracy could be obtained using state-of-the-art DNN architectures such as CNN, LSTM or TCN.

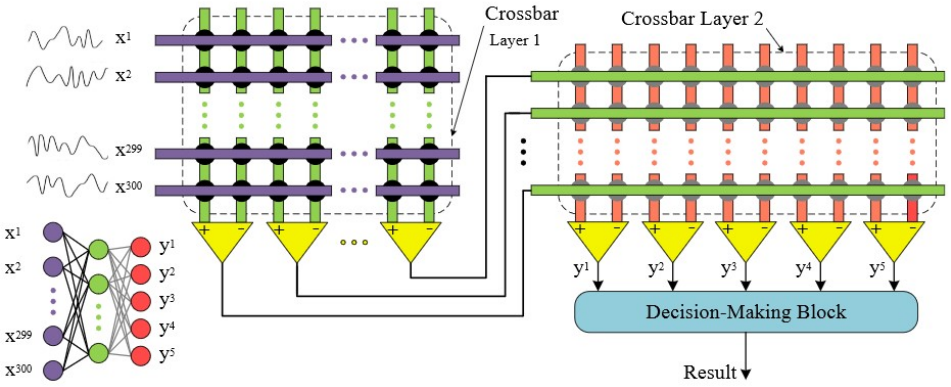

Figure 3.3: Hardware implementation of the two layers memristor crossbar array used in [28]

3.3. Long Short Term Memory Network (LSTM) and Bidirectional LSTM

Qiao *et. al.* [47] propose a deep extreme learning machine based on local receptive field (DELM-LRF) BLSTM model for a fast and efficient ECG signal recognition. Different from the

common understanding that hidden neurons of neural networks need to be iteratively adjusted during training stage, ELM theories show that hidden neurons are important but need not be iteratively tuned. The neurons are initialised randomly and a method of pseudo matrix inverse is used to analytically calculate the output weight matrix using least square algorithm. The first stage consisting of DELM-LRF is used to learn the significant spatial features. The network architecture is shown in Figure 3.4. In the feature extraction stage, convolution and max pooling layers are alternately stacked to extract spatial features. Following which, the features are fed to a three layer BLSTM to learn the temporal features and perform a classification. It has a fast training time because of reduced features. The intermediate layers have randomized initialized weights whereas only the final layer is trained for the weights. The classification algorithm has low computational complexity and high performance making it suitable for hardware implementation. The ELM training rule is applicable for a single layer of weights, while other weights were initialised (and used) randomly. They obtain a high accuracy of 99.32%, F1-score 97.1%. However, the architecture in [47] uses additional layers for feature extraction for LSTM classification layers. Use of large network size will consume high energy in hardware.

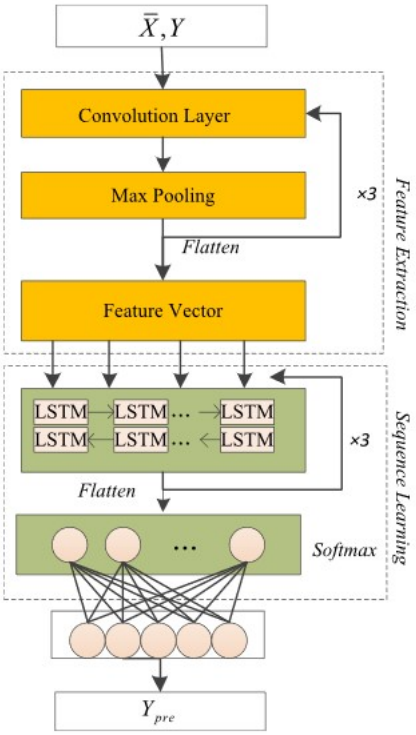

Figure 3.4: Extreme Learning Machine based on Local Receptive Field BSLTM architecture used in [47]

Chauhan et. al. [14] propose a simple network topology based on LSTM network where a small network using two stacked LSTM layers are used to classify the ECG signal. They follow an approach of not detecting the 'QRS' peaks of ECG signal and instead slicing the signal. The accuracy obtained is 96.45% and F1 score 93.88%. Even though a high accuracy is obtained, they have addressed only a subset of arrhythmia classes. In case of hardware implementation, the solution will be suitable for low power applications.

In the work by Ozal *et. al.* [60], a high accuracy ECG arrhythmia classification using deep bidirectional LSTM (BLSTM) network is proposed. A wavelet sequence (WS) layer is added to the network which is used for feature extraction and the wavelet coefficients are fed to

the neural network along with the main signal. The wavelet sequence layer uses a series of low pass and high pass filters. The network architecture is shown in Figure 3.5. After the wavelet sequence layer; Unidirectional LSTM and BLSTM layers are placed. They obtain a high accuracy of 99.39% with the combination BLSTM+WS. Another usage of bidirectional LSTM is demonstrated by Acharya *et. al.* [41]. They suggest an automated detection method of a specific type of arrhythmia - Atrial Fibrillation (AF) in ECG signals. They use bidirectional LSTM followed by a fully connected layer to detect AF. MIT-BIH Atrial Fibrillation database is used in the work. They obtain a high accuracy of 98.51%. But only a subset of arrhythmia classes have been addressed in both the works, which does not validate the high accuracy to its actual potential.

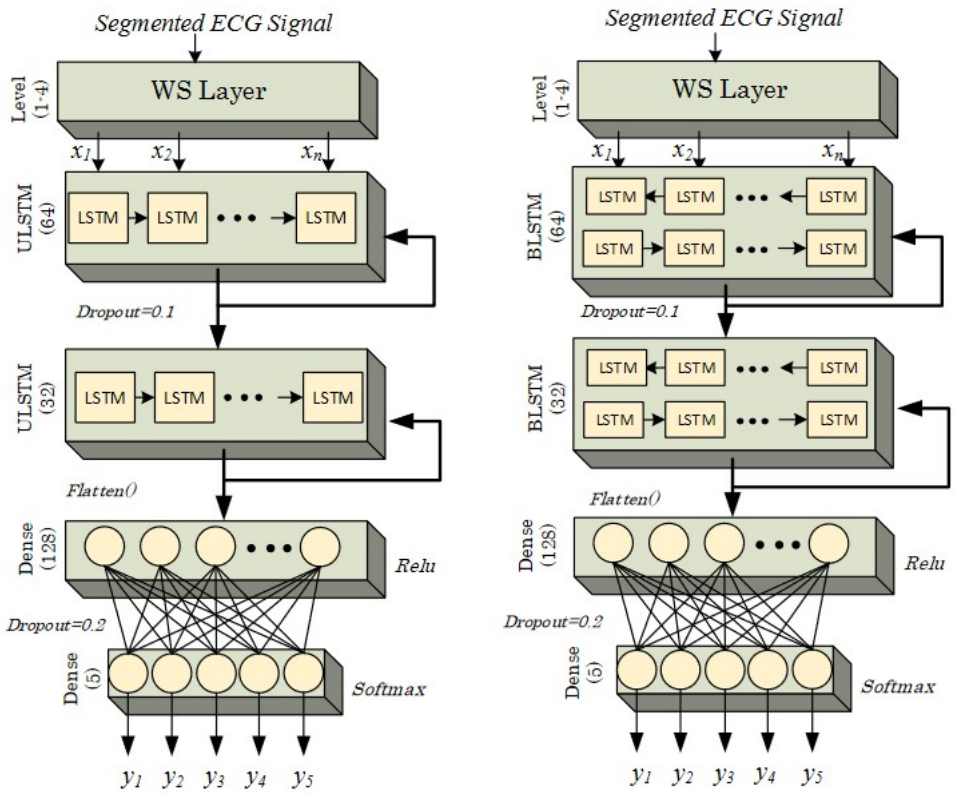

Figure 3.5: Structure of the unidirectional and bidirectional LSTM network used in [60]

3.4. Hybrid of CNN and LSTM

Combination of LSTM layers with CNN is also demonstrated in the literature. Ping *et. al.* [63] combine CNN and LSTM in order to obtain a model for generalizing the trained network to new patients. The adopted network architecture as shown in Figure 3.6 uses thirteen 1D convolutional layers in combination with one LSTM layer. They have considered five global classes N, SVEB, VEB, F, Q in the work. Before picking the LSTM + CNN model, a comparison between SVM, LSTM, CNN and LSTM + CNN is done. Accuracy of 88.38% is obtained in their proposed model paradigm and accuracy of 99.08% is obtained with the conventional testing paradigm i.e. test the network for the patients that the network is already trained with. The work does not adopt a finer level beat-by-beat classification and no QRS peak detection is done. They also use a large neural network which consumes higher energy and higher processing time.

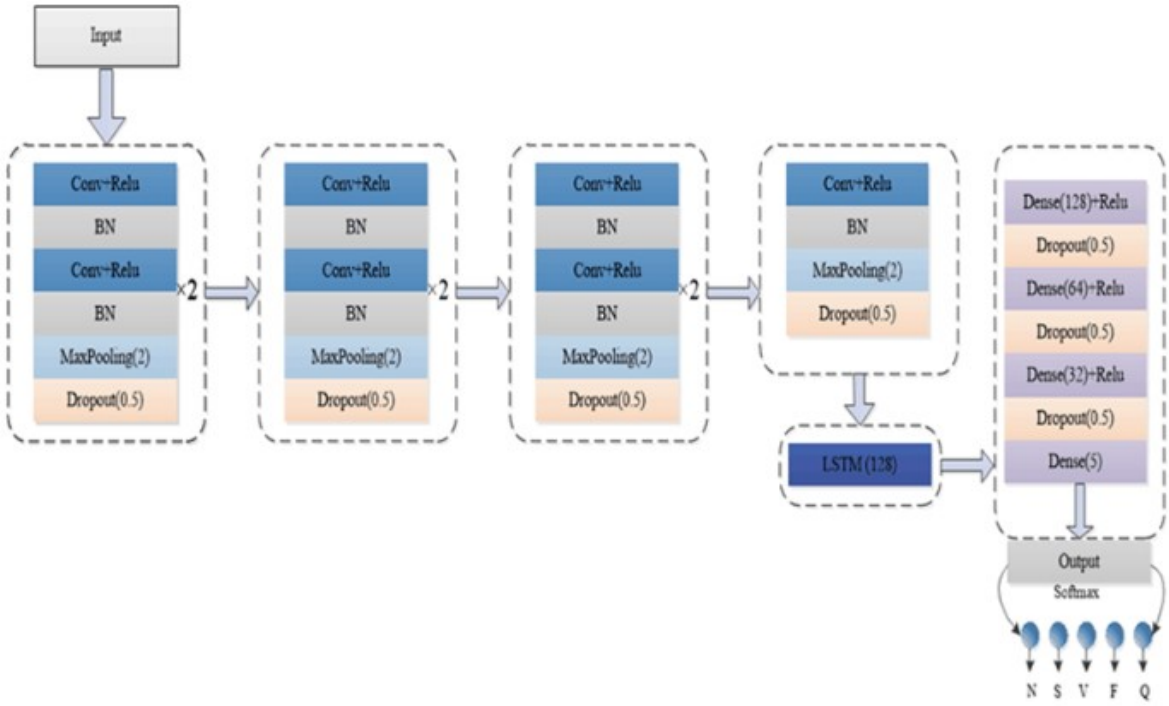

Figure 3.6: Network architecture as a combination of CNN and LSTM used in [63]

A similar approach of combining CNN and LSTM is done by Panga *et. al.* [53]. The algorithm slices a 5s ECG segment into 211 short segments of 24 data points per segment. These are fed into the CNN layer to perform feature extraction before going into the LSTM layer. However, they perform only a binary classification. Accuracy of 99.8% is obtained. The time dependency between the samples of ECG signal was learned using the LSTM layer. Since both CNN and LSTM models learn different functions, merging these two nets yielded a higher classification accuracy [53].

In another work, Ozal *et. al.* [62] use deep coded features using Convolutional Auto Encoders (CAE) for classifying ECG signals using LSTM network. Their network architecture is shown in Figure 3.7. The auto encoder is based on non-linear compression and reduces the signal size of arrhythmic beats. It not only reduces complexity of computation by the classifier model but also reduces the time for training due to smaller lengths. It is beneficial for wearable technologies as they can store longer duration of ECG signal consuming less power. High accuracy of 99.0% is obtained. The solution also allows efficient transmission of signal over-the-air or wired channels due to reduced feature size. However, the solution requires additional processing of raw signal to obtain the coded features. Besides, only a subset of arrhythmia classes is considered for classification which does not justify the high accuracy for practical usage.

Instead of using a deep network for reducing feature set, another instance of dimensionality reduction would be to use Linear Discriminant Analysis (LDA). Lee *et. al.* [33] use LDA for reducing the feature vector of ECG arrhythmia. Input features of ECG signal obtained by wavelet transform is reduced from 17 to 4. They show that LDA performs better than Principal Component Analysis (PCA) for dimensionality reduction. Accuracy of 99.48% is obtained. Using a reduced feature set results in faster computation speed and cuts down the computational costs dramatically.

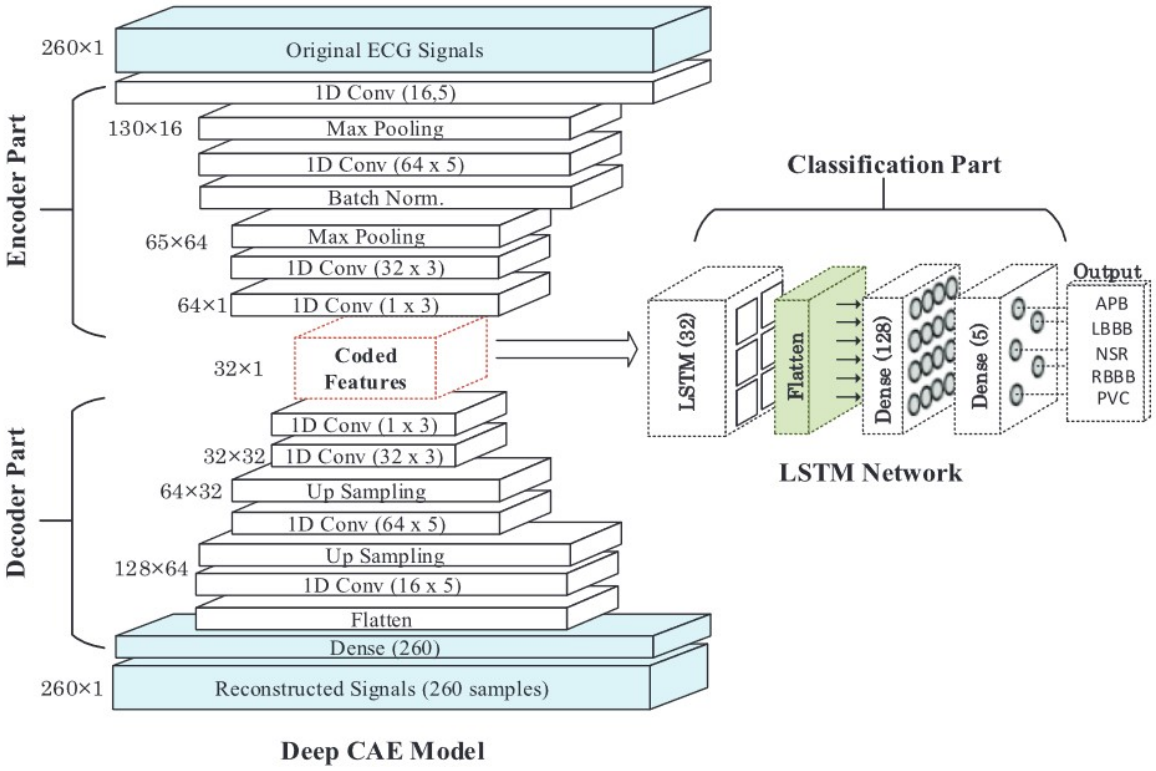

Figure 3.7: Convolutional Auto Encoders with LSTM for classification used in [62]

3.5. Temporal Convolutional Network (TCN)

Ingolfson *et. al.* [37] propose a 'ECG_TCN' architecture using TCN network for ECG arrhythmia classification on hardware. The network architecture consisting of four TCN layers in addition to a convolutional layer is shown in Figure 3.8. They test their implementations on the STM32L475, which is based on ARM Cortex M4F, and the GreenWaves Technologies GAP8 on the GAPuino board. Instead of MIT-BIH, the dataset called ECG5000 is used in the work. Compared to ARM Cortex M4F, GAP8 is 23x more energy-efficient and 46.85x faster. They classify ECG beats into five output classes having accuracy of 94% and consume 0.1 mJ of energy during inference. However, compared to other works, they have less accuracy and also high energy is consumed in hardware implementation.

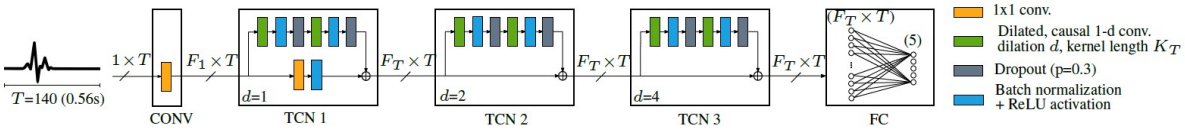

Figure 3.8: ECG_TCN network architecture used in [37]

In another work, Yang *et. al.* [59] have used TCN network for anomaly detection in several time-series sequences including ECG signal. They classify the ECG signal input into binary classes - normal and abnormal. In order to achieve better information processing, the network architecture is modified to include concatenated outputs from multiple hidden layers at the classification layer. F1 score of 90% is obtained for the model. They use a unique unsupervised classification technique where network is trained only for normal beats and identifies abnormal beats during inference. However, classifying into more number of classes could validate the performance of the model better.

3.6. Spiking Neural Network (SNN)

Spiking Neural Network (SNN) have recently emerged as a prominent neural computing paradigm. They are more biologically plausible and incorporate the aspect of time. The neurons model the synaptic state, and unlike other artificial neural networks (ANN), the information between layers is passed only when the neuron fires. The input signal is first encoded into a spike train using an encoding scheme. The neuron is usually modelled as leaky integrate and fire (LIF) model, as shown in Figure 3.9. The membrane potential of a neuron increases with the incoming spike current. As and when the membrane potential exceeds a threshold, the neuron spikes. Then the membrane potential of the neuron is reset. Decoding methods for interpreting the outgoing spike train as a real-value number, rely on either the frequency of spikes (rate-code) or the interval between spikes.

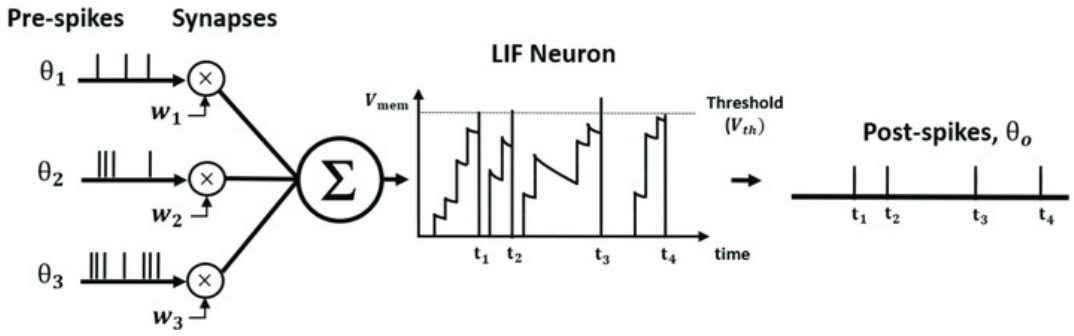

Figure 3.9: An illustration of a Leaky Integrate and Fire Neuron [32]

A variation of recurrent neural network (RNN) using spiking neurons known as Spiking RNN (sRNN) is proposed by Bauer *et. al.* [12]. It is demonstrated on a DYNAP-SE development kit which is a mixed signal neuromorphic processor for detecting anomalies in ECG and its like signals, developed by ETH Zurich. ECG signals are converted to trains of events through a sigma-delta encoding scheme. The spike train is fed to two layers of neurons - input expansion and a second layer of excitory and inhibitory neurons. The output of the second layer is also a spike train. In the second output layer, also called the readout layer, if the weighted sum of spiking activities exceeds a threshold, anomaly is detected. The weights are trained using supervised method. The weights of the recurrent network are initialized randomly and learning or weight update takes place only in the output layer, which reads out the state of a hidden layer sampled from the recurrently connected neurons. Thresholds are chosen such that the number of false negatives for the respective anomaly and the number of false positives are minimised over the validation set. They obtain a overall specificity of 97.6%. They have proposed an innovative concept of modifying RNN neurons with spiking neurons suitable for low power applications. However, only anomaly detection is done in the work. Besides, deciding thresholds for the neuron to spike is difficult for getting the optimal accuracy level.

Amirshahi *et. al.* [9] demonstrate ECG signal classification using SNN based on Spike-timing-dependent plasticity (STDP) and reward based STDP (R-STDP) learning. Their model consists of multiple layers including spike encoding, Gaussian layer, STDP layer, inhibitory layer and R-STDP layer as shown in Figure 3.10. The spikes are generated from the ECG beat with overlapping windows, where each window spans a part of the beat. The spike is generated based on a LIF neuron. The model weights are trained according to the timings of spike signals. A new concept of reward and punishment for weights is proposed, reward - if prediction is correct and punishment - if prediction is incorrect. Unlike STDP, R-STDP is a supervised learning algorithm. It operates not only based on spike timings but also a modulator signal. They cover four global classes: N, VEB, F and Q. The proposed solution is

implemented based on Brian2, a Python package for evaluation of spiking neural networks. They obtain an accuracy of 97.9%. However, they adopt a patient specific training which requires training the network every time for a new patient and limits the performance of the network.

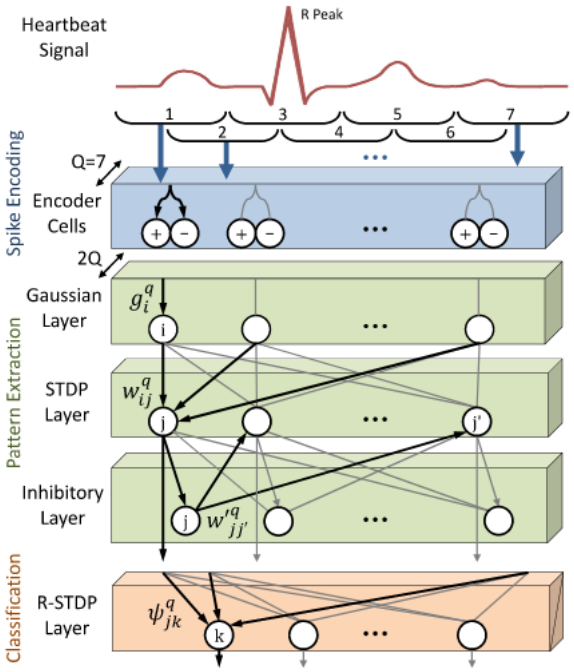

Figure 3.10: Network architecture proposed in [9]

Corradi *et. al.* [16], implement ECG signal classification on a custom event-driven mixed-signal neuromorphic processor as well as software. A recurrent SNN is used as the classifier model. Instead of using a single spike train, a pair of spike trains are used for representing positive and negative amplitudes. Training of the neurons in software is carried out by first training a SVM where the objective is to find a hyperplane that separates the output classes. The decision boundary is obtained by calculating the dot-product between input test pattern and weight connections. This decision boundary forms the membrane threshold potential of the neuron which spikes upon crossing threshold. The points in the SVM hyperplane is modelled as weights for neurons which solve a classification problem. A unique method of converting weights obtained from SVM training into recurrent SNN weights is done which is suitable for low power applications. However, a lower accuracy of 95.6% is obtained. Besides, training of SNN is not done directly which could have provided a better comparison between the two approaches.

Comparative Study: Artificial Neural Networks and Spiking Neural Networks

Based on the above network analysis, most of the SNNs have performed poorly in terms of accuracy compared to their ANN counterparts. Nonetheless, SNNs consume much lower power compared to ANNs. A detailed study comparing ANN and SNN has been performed for standard datasets such as MNIST and CIFAR-10 by Lei *et. al.* [18] and similar arguments can apply for ECG arrhythmia classification.

- Data Precision of ANNs is much higher than SNNs, which makes ANNs easier to gain better recognition accuracy with the same network size [18]. The reason is that there

is information loss when converting the original image/signal to a spike signal for consequent SNN processing. However, on spike-oriented workloads such as DVS-CIFAR, SNN achieves better accuracy. But at present all datasets are ANN oriented and there is lack of any benchmark dataset for SNN. Instead, ANN oriented static datasets are being modelled with spiking inputs leading to loss of information. Therefore, the current implementations can only be seen as a proof-of-concept for spiking networks [18].

- ANNs have a greater number of multiply and accumulate (MAC) computations and hence consume higher power. Since SNNs deal with only spikes (binary), addition is the only main computation overhead, making SNNs consume lesser power.
- The training algorithm for ANNs use back propagation. Back propagation is a standard training algorithm for all feed forward neural networks and is easy to use. Spike Timing Dependent Plasticity (STDP) is the training algorithm commonly used for SNNs. However, there is no training algorithm for SNNs which has reached the same level of popularity as back propagation. The training algorithms for SNNs are difficult to design and analyze because of the asynchronous and discontinuous way of computing ([44], [18]).

Therefore, considering accuracy and the nature of available dataset, ANNs are chosen for this thesis. Then, optimization of energy is targetted.

3.7. Summary

State-of-the-art works using deep learning methods to perform ECG arrhythmia classification has been reviewed in this chapter. Few common challenges still exist in these works which can be further improved.

- There is a lack of focus on performing classification for all ECG arrhythmia classes and only a subset of arrhythmia classes have been considered in most of the works ([61], [60], [62], [35], [14], [12], [28], [53]). In such a case, since the model is exposed to only a small number of classes, the network learns the distinction between classes much better. This also gives a higher accuracy. However, it may not be a suitable approach to validate the neural network model to its true potential. If the patient is using an ECG device which is capable of detecting only a subset of arrhythmia classes, it is not dependable for practical medical usage. Besides, going for a finer level classification is equally important. Apart from disfunctionality of the heart itself, arrhythmia can be caused by several other problems like high blood pressure events, side effects of certain medications, surgery etc. Knowing the exact heart arrhythmia that the patient is suffering from, can help isolate or track down the cause of the arrhythmia better and speed up the treatment process in the right direction. Arrhythmia treatment includes anti-arrhythmic drugs, anti-coagulants, use of pacemakers, catheter ablation etc.
- Many works have adopted neural networks with large number of layers to obtain high accuracy ([56], [42], [61], [63], [62], [47]). Using large network layers requires more number of weights to be trained. This approach will not only take higher processing time but will also consume higher energy since large crossbar arrays and more periphery devices will be required for hardware implementation.
- For hardware level solutions, it is difficult to satisfy both high accuracy/F1 score and low energy requirement simultaneously. SNNs require low energy, but they have less accuracy/F1 score ([12], [16]). While large CNNs and CNN+LSTMs have high accuracy but consume high energy ([56]).

Therefore, in this thesis, the challenges of low accuracy and high energy consumption is addressed to design a complete ECG classification solution for all arrhythmia classes using RRAM based DNN framework.

4

Proposed Scheme

The proposed scheme is discussed in detail in this chapter. Section 4.1 presents the overview of the proposed ECG classification scheme. Section 4.2 introduces the model of hierarchical three stage DNN. Section 4.3 briefs on the RRAM based DNN architecture for the proposed scheme. DNN topology exploration is described in Section 4.4. Section 4.5 discusses the energy optimization with dimensionality reduction followed by Section 4.6 detailing upon accuracy optimization with partial training for the personalization. The complete hierarchical three stage DNN architecture is presented in Section 4.7.

4.1. Proposed ECG Classification Scheme

As we saw in previous chapters, deep learning using Deep Neural Networks (DNN) have several advantages over traditional artificial intelligence (AI) techniques. They have the skill to learn features automatically, eliminating the need for manual feature extraction of inputs. For medical applications such as ECG arrhythmia classification, DNN enables accurate health-data interpretation and a timely clinical action by reducing human intervention. In this thesis, a complete solution for ECG arrhythmia classification is proposed using a hierarchical three stage DNN, targeting high accuracy and energy-efficiency using resistive random access memory (RRAM) based DNN architecture.

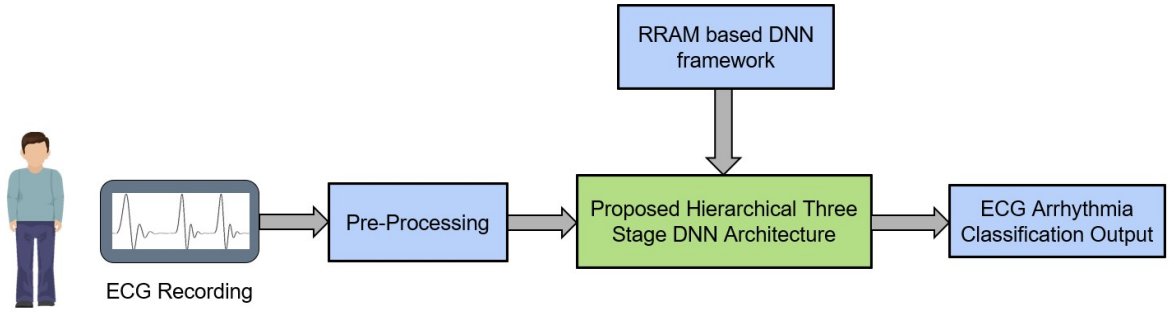

Figure 4.1: Overview of the proposed ECG classification scheme

An overview of the proposed ECG classification scheme is shown in Figure 4.1. The ECG signal is recorded using sensors attached to the human body. To ensure an error-free classification, it is recommended that the person be in a stable physical condition or the physiological condition of the human body should be at rest. The ECG signal is then processed to extract the ECG beats. After pre-processing, the ECG beat is ready to be used by the DNN for arrhythmia

classification.

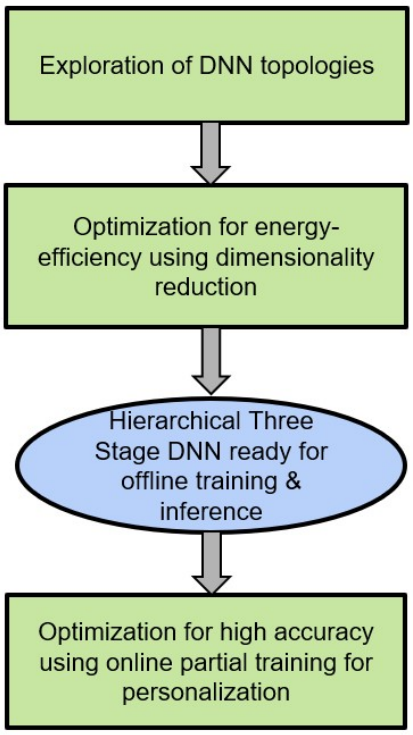

Figure 4.2: Architecture development steps

The architecture development steps for the ECG classification scheme is shown in Figure 4.2. To pick the best DNN topology for the proposed scheme, exploration of various state-of-the-art neural networks is carried out and extensively evaluated for their performance metrics trade-offs. The most efficient DNN topology in terms of accuracy and energy is chosen for the hierarchical three stage DNN architecture.

Energy optimization of the chosen DNN topology is done using dimensionality reduction techniques. After this step, the hierarchical three stage DNN is ready for offline training and arrhythmia classification. Further, accuracy optimization is achieved for personalization using online partial training. In further sections, a model of hierarchical three stage DNN is described followed by each of the architecture development steps.

4.2. Model of Hierarchical Three Stage DNN

The ECG dataset follows an uneven class distribution. The abnormal beats are present in very less number compared to the normal beats. Due to the split of the dataset and the nature of data distribution, a hierarchical three stage DNN architecture has been proposed in this work. The schematic of the proposed architecture is shown in Figure 4.3. The three stages of the network architecture classify the ECG beats at different hierarchical levels based on Table 2.1. Using selective network activation, the full DNN scheme performs an energy-efficient ECG classification.

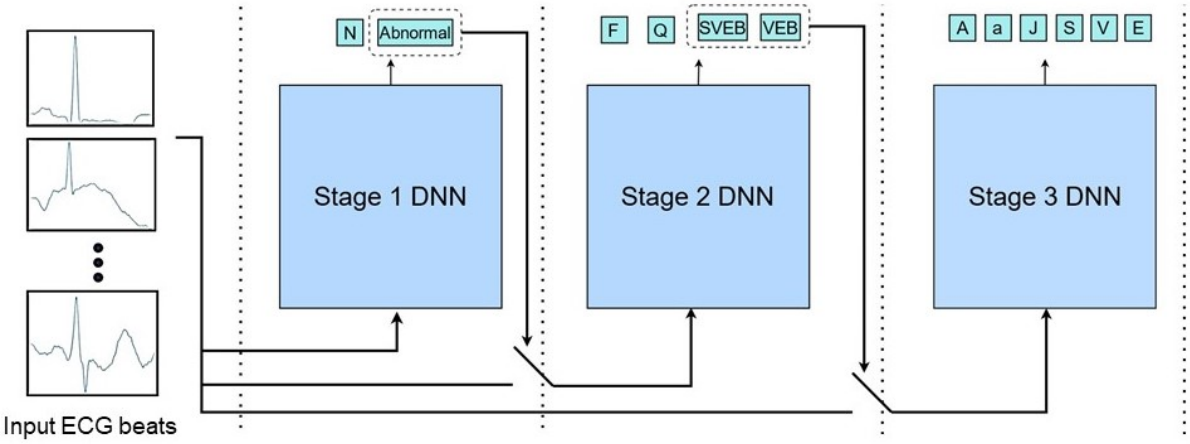

Figure 4.3: Schematic of Hierarchical Three Stage DNN architecture to classify input ECG beats at three different hierarchical levels

Table 4.1: Purpose and energy-efficiency of each stage of the heirarchical three stage DNN model

Stage	Purpose	Energy-efficiency
Stage 1	This stage is used to classify the input ECG beats into Normal (N) and abnormal classes. When abnormal beats are detected by Stage 1, it is sent to Stage 2 for further classification.	This stage is operated 100% of the time for all beats. Since most of the ECG beats are 'Normal', a majority of the network remains deactivated for most of the time. This brings high energy saving to the system. A simple network topology, using fully connected network is used at this stage as shown in Figure 4.4. Utilising a resource efficient fully connected network for Stage 1, which is operated 100% of the times, ensures less energy consumption.
Stage 2	This stage is used to classify abnormal beats into its sub-categories which include Ventricular Ectopic Beat (VEB), Supraventricular Ectopic Beat (SVEB), Unknown Beat (Q) and Fusion Beat (F)	This stage is turned on only when an abnormal beat is detected, for nearly 17.6% of the cases. Based on the classification output of Stage 2, the next stage of the network is activated or else it remains deactivated. When detecting any VEB or any SVEB beats, Stage 2 beats are processed by Stage 3. Therefore, Q and F beats which do not require any further classification are left at this stage, reducing energy consumption.
Stage 3	This stage classifies the VEB or SVEB beats into their sub-categories Atrial Premature Beat (A), Aberrated Atrial Premature Beat (a), Nodal (junctional) Premature Beat (J), Supraventricular Premature Beat (S), Premature Ventricular Contraction (V) and Ventricular Escape Beat (E).	The beats filtered by Stage 2 reach Stage 3. Therefore, this stage is turned on only when a fine level classification is needed for approximately 9.2% of the cases.

Stage 2 and 3 require more complex DNN networks to ensure high accuracy and energy-efficiency. Therefore, several state-of-the-art neural networks were explored and evaluated for their performance metrics trade-off. The DNN topology exploration is presented in Section 4.4.

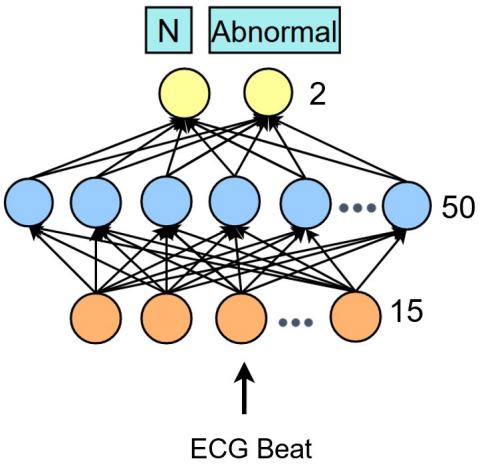

Figure 4.4: Stage 1 DNN topology utilising a resource efficient fully connected network to ensure less energy consumption

4.3. RRAM based DNN framework

The RRAM based DNN framework considered in this work is based on the computation in-memory (CIM) architecture proposed in ISAAC [51]. The CIM architecture uses the memristive devices arranged in a crossbar array structure to perform computations. The fundamental operation of all the neural network computations is a Multiply and Accumulate (MAC) operation, represented using Vector Matrix Multiplication (VMM). The resistive state or the conductance level of the RRAM is modified to represent network weights, by applying voltages and currents. However, the bit-precision of memristor cells are usually less than that required by the neural network weights. Therefore, bit-slicing technique is utilized in ISAAC [51], where the bits are divided into smaller chunks. The bit-sliced network weights are mapped to the memristor conductances. The ISAAC architecture [51] considers a 16-bit resolution for input voltages and network weights. The 16-bit weights are mapped onto group of 8 RRAMs, with each RRAM programmed with a 2 bit-slice as shown in Figure 2.18. Each 16-bit input is converted to analog voltage and given to the crossbar with 1-bit at a time.

Figure 2.18 in Section 2.4.2 shows the mapping of a VMM operation onto a memristor crossbar array structure, with each memory element in the cross section of vertical and horizontal rails. It shows the crossbar structure for a single weight matrix. Multiple crossbars are needed to represent different weight matrices for the hierarchical three stage DNN model. The size of weight matrices determine the size of crossbar array. Larger crossbar arrays also require additional periphery devices, leading to more energy consumption. With selective network activation and dimensionality reduction of the hierarchical three stage DNN, the energy consumption of the crossbar arrays and the associated periphery devices gets reduced.

4.4. Exploration of DNN Topologies

A comprehensive evaluation of DNN topologies was carried out to select the most suitable network topology for Stage 2 and Stage 3 of the hierarchical three stage DNN architecture.

For the exploration of the DNN topology, neural networks such as Long-Short-Term-Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN), and the Hybrid architecture of TCN and LSTM were considered. Output classes for classification included Normal (N), Supraventricular Ectopic Beat (SVEB), Ventricular Ectopic Beat (VEB), Fusion Beat (F) and Unknown Beat (Q). The DNN topologies for classifying ECG beats into the five output classes are shown in Figure 4.5. A detailed description of network topologies itself can be found in Section 2.2.

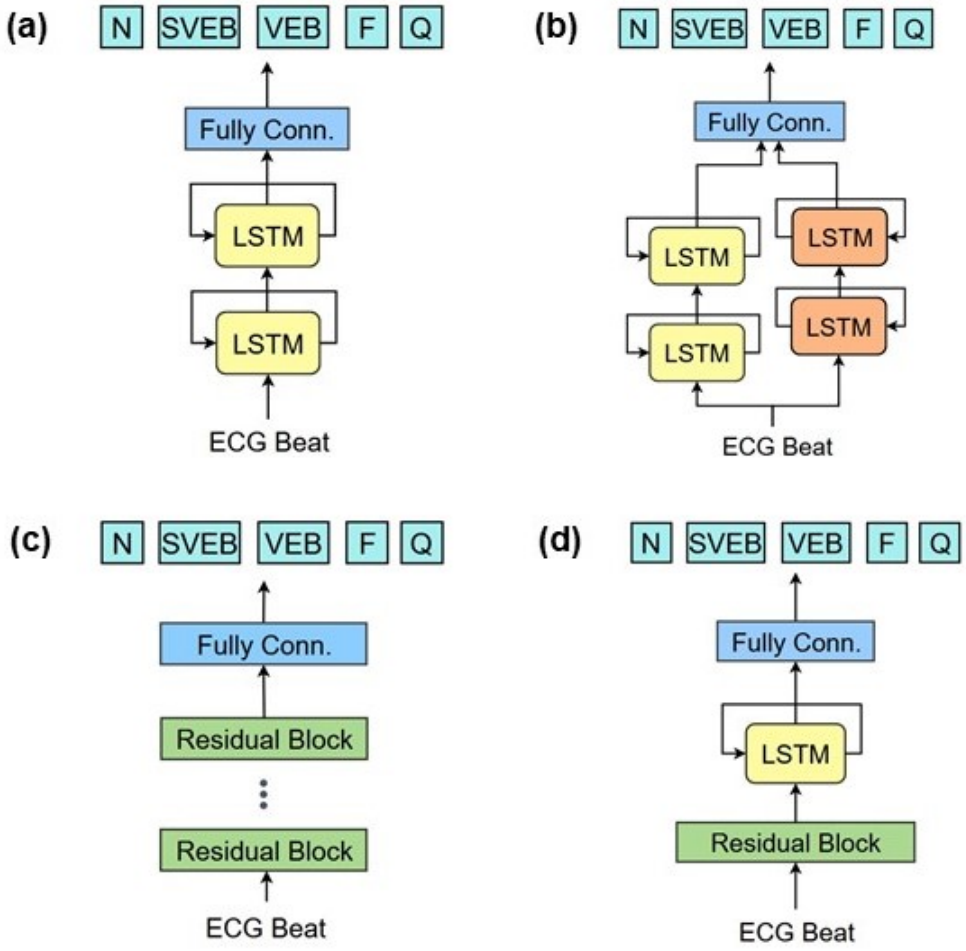

Figure 4.5: DNN topologies (a) LSTM (b) BLSTM (c) TCN (d) Hybrid of TCN & LSTM

A qualitative comparison between the DNN topologies is shown in Figure 4.6. As already discussed, BLSTM uses two copies of LSTM networks and processes the inputs in two directions. Hence, BLSTM network has a higher energy and area requirement. Also, due to more and better information processing, BLSTM has a high accuracy compared to LSTM. This also makes BLSTM slower than LSTM.

To match the size of the input receptive field, TCN requires many residual block layers stacked on top of each other. This is in contrast to lesser layer requirement for LSTM and BLSTM. Besides, while LSTM and BLSTM store information in their hidden state during inference, TCN requires the inputs to be available at all times. Thus, TCN may have a higher memory requirement [11]. On the other hand, while LSTM and BLSTM require additional hardware such as pointwise multipliers for output computation, TCNs do not require additional hardware and completely depend upon RRAM crossbars for their heavy computational tasks like convolutions. Therefore, for equal number of layers, TCN will have lesser area requirement than

LSTM or BLSTM. But since large number of TCN layers are required for ECG classification, TCN have higher area, energy requirement compared to other DNN topologies. To overcome this, a hybrid architecture is also explored in this work. Few TCN layers are replaced with LSTM layers for obtaining reduced area and energy consumption. Hybrid network also provides faster processing for inference of ECG beat than TCN. Even though hybrid architecture provides a reduced area and energy consumption, there was not much gain with accuracy. Hence, Hybrid was not selected for Stage 2 and Stage 3 DNN topologies. Based upon the requirement of high accuracy and energy-efficiency, performance trade-offs between various architectures was done to pick LSTM as the most suitable DNN topology for Stage 2 and Stage 3. (Please refer Section 5.2 for evaluation results.)

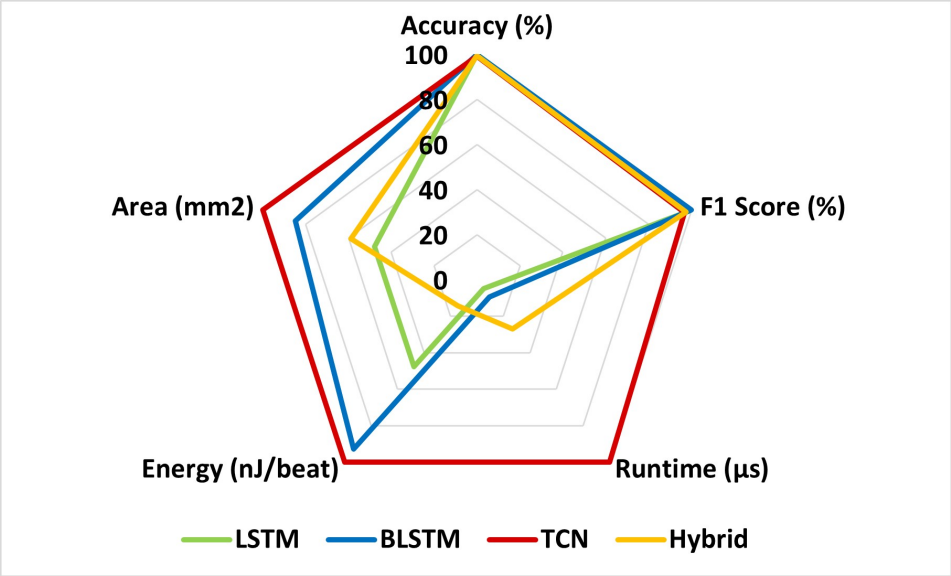

Figure 4.6: Qualitative comparison of various DNN topologies (please refer Section 5.2 for detailed results analysis). BLSTM has highest accuracy/F1 while LSTM has the most optimal performance.

4.5. Optimization using Dimensionality Reduction

The topology selected from the DNN exploration for the hierarchical three stage DNN architecture is optimized for attaining energy-efficiency using dimensionality reduction.

Reducing the amount of information in the input data leads to reduced processing time, reduced computational complexity and hence, reduced energy consumption. Therefore, prior to doing classification, dimensionality reduction is applied on the input data. Transforming the dataset by reducing the number of variables may bring down accuracy levels. But the idea is to apply dimensionality reduction techniques with minimal or no loss in accuracy, while still achieving energy-efficiency. Dimensionality reduction techniques adopted in this work are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Discrete Wavelet Transform (DWT). DWT reduces a ECG beat of 250 samples to 25 samples, PCA reduces to 15 samples and LDA to 4 samples. Each of the reduction techniques is described in detail in Section 2.3.

Applying dimensionality reduction on input data reduces the size of network parameters, and reduces energy consumption. Figure 4.7 shows the impact of dimensionality reduction on network parameters. The number of input neurons in a fully connected network depends on

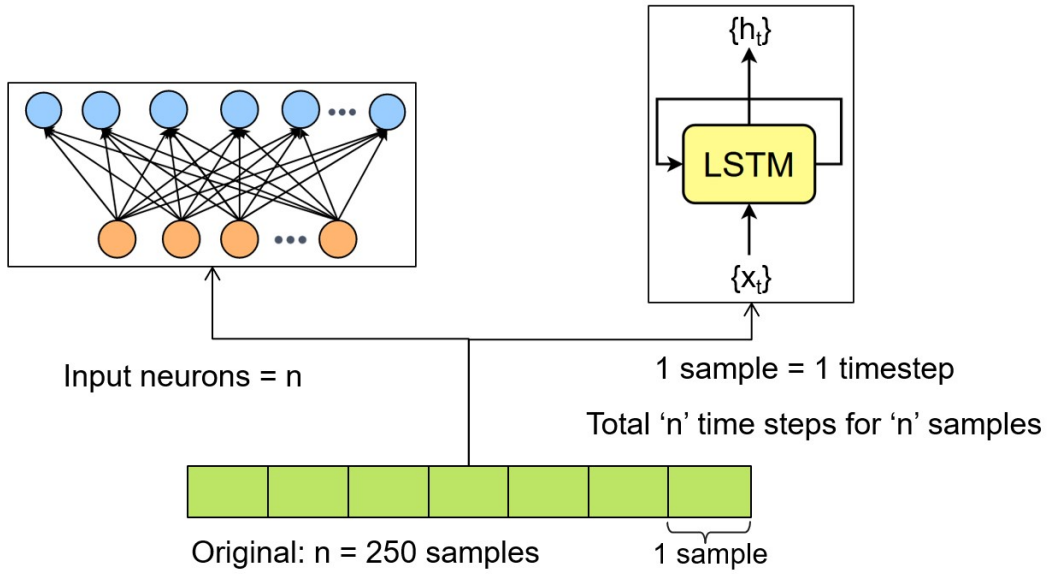

Figure 4.7: Impact of dimensionality reduction on network parameters

the size of input. When a full ECG beat of length 250 is given to a fully connected network, 250 neurons are required for the input layer. However, if the ECG beat is reduced to, for example 15 samples using PCA, the number input neurons also reduces to 15. This brings down the size of weight matrix and hence, energy reduction is obtained. Similarly, the number of timesteps to give input to the LSTM network depends upon the length of input. Since one sample is given at one timestep, reducing the total number of samples reduces the total number of timesteps to give the input. Hence, the number of network traversals reduces and energy-efficiency is obtained.

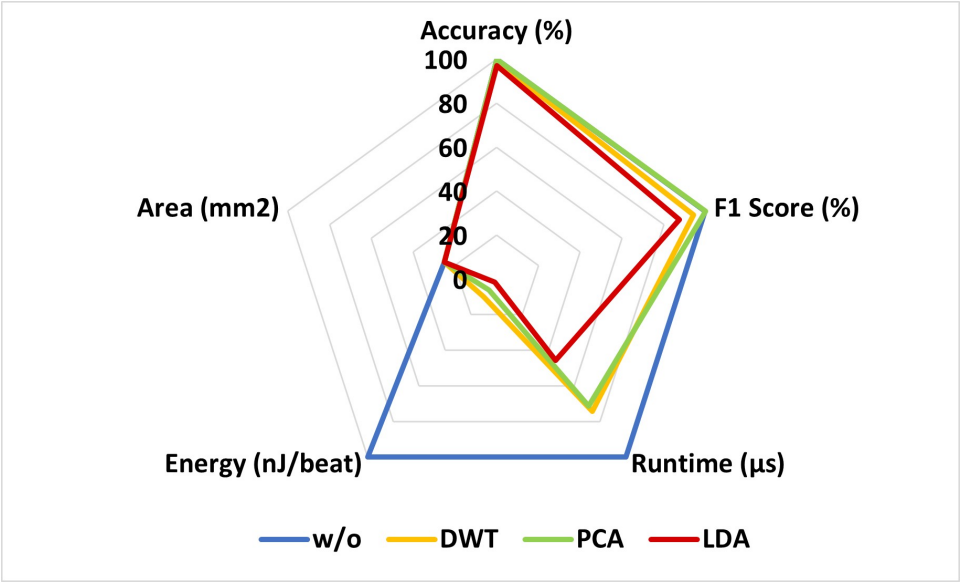

Figure 4.8: Qualitative comparison of various dimensionality reduction techniques (please refer Section 5.3 for a detailed analysis). PCA with highest accuracy/F1 has the best performance.

A comparison between dimensionality reduction techniques is shown in Figure 4.8. LDA

provides the maximum energy reduction, followed by PCA and DWT. Even though maximum energy efficiency is obtained with LDA, the classification accuracy of LDA is least compared to others. Therefore, based upon the requirement of high accuracy and energy-efficiency, performance trade-offs between various techniques was done to pick PCA as the most suitable the dimensionality reduction technique for the hierarchical three stage DNN architecture. (Please refer Section 5.3 for evaluation results.)

4.6. Optimization using online Partial Training

Personalization of medical devices is essential because biomedical signals such as ECG are unique to each patient and require additional training to accurately classify individual users' beats. An online personalised training approach to regain device accuracy is proposed in this work. Once the device is fully trained offline for all patients, the network weights are ready to be programmed onto the RRAM devices. But when the device is deployed on-field, it may be used for a new person. Since ECG data of the new user was not used for offline training, the accuracy of the network may not be high. Therefore, there is a need to go for online personalized training to regain device accuracy. This requires updating or fine-tuning the network weights to adapt to the new person. This online personalized training can be done by either full network training or partial training. This is shown in Figure 4.9. Full network training is a standard training technique which is costly both in terms of time and energy consumption. Therefore, energy-efficient online partial training technique is used in this work.

Instead of training the entire network which includes modifying weights for both the LSTM layers and fully connected layers,

Since fully connected layers are inexpensive in terms of energy consumption, partial training provides a significant amount of energy saving. Besides, since RRAM devices suffer from endurance problem, modifying the network weights with the partial training approach reduces the endurance problem.

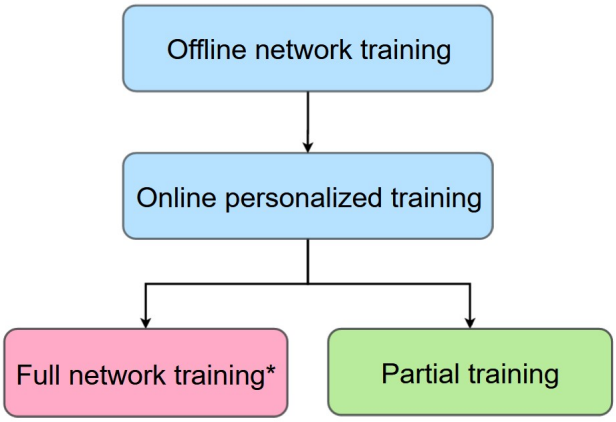

Figure 4.9: Training for personalization.
∗Full network training is a standard patient-specific training choice [9].

4.7. Complete Hierarchical Three Stage DNN Architecture

The complete hierarchical three stage DNN architecture is shown in Figure 4.10. Supervised training is adopted in this work, where the set of ECG beats along with class labels are given to the network for training. The network takes a set of ECG beats and classifies it into their corresponding output class. The three stages of the network model classify the ECG beats

at different hierarchical levels based on Table 2.1. Stage 1 fully connected network classifies the input ECG beats into Normal (N) and Abnormal classes. If Abnormal beats are detected in Stage 1, it is processed by the Stage 2 LSTM network. Stage 2 classifies the beats into Fusion Beat (F), Unknown Beat (Q), Supraventricular Ectopic Beat (SVEB), and Ventricular Ectopic Beat (VEB). Since SVEB and VEB require further subcategory classification, they are sent to Stage 3, while F and Q beats are dropped at Stage 2. Stage 3 LSTM network classifies the beats into Atrial Premature Beat (A), Aberrated Atrial Premature Beat (a), Nodal (junctional) Premature Beat (J), Supraventricular Premature Beat (S), Premature Ventricular Contraction (V) and Ventricular Escape Beat (E). To obtain energy-efficiency, dimensionality reduction is applied on the input ECG beats before giving the beats to the network. Input ECG beats are reduced with principal component analysis (PCA) in this work. PCA was chosen as it is the most efficient dimensionality reduction technique in terms of accuracy and energy efficiency. After this step, the hierarchical three stage DNN is ready for offline training and arrhythmia classification. Further, accuracy optimization is achieved for personalization using partial training. For device personalization, the weights of the last fully connected layer of each stage is trained online with energy-efficient partial training approach for regaining high accuracy.

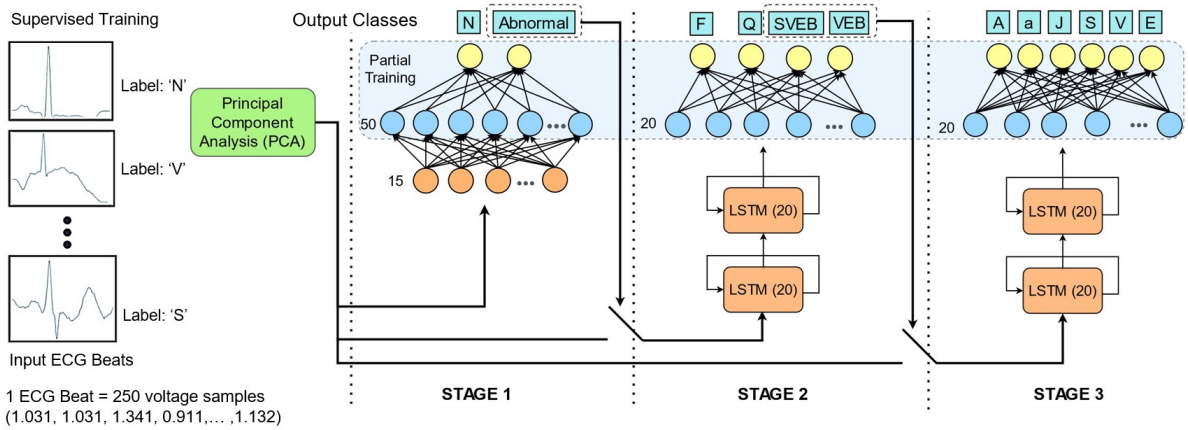

Figure 4.10: Complete Hierarchical Three Stage DNN Architecture to classify input ECG beats into hierarchical output classes. Uses dimensionality reduction for energy optimization and partial online training for accuracy optimization.

5

Simulation Results

This chapter presents the details of the simulation results. Section 5.1 gives details of the simulation setup. Section 5.2 presents the results of the DNN topology exploration. Section 5.3 gives the simulation results for dimensionality reduction, followed by obtained results for hierarchical three stage DNN in Section 5.4. Section 5.5 presents the partial training results. Finally, Section 5.6 compares the proposed scheme with state-of-the-art.

5.1. Simulation Setup

The simulation setup used in the work including the dataset, software and hardware setup is described below:

5.1.1. Dataset

The dataset considered in the thesis is the MIT-BIH arrhythmia database [40], available Open Source in Physiobank. The database consists of 30 minutes of ECG recording from 48 individual patients. It consists of ECG beats from fifteen arrhythmia types. The distribution of ECG beats for each of the arrhythmia type is given in Table 5.1. The dataset consists of total 104,417 beats. A split of 80%-20% is used for training and testing set, respectively.

5.1.2. Software Setup

PyTorch [8] is an open source machine learning library which allows rapid prototyping of deep neural networks. It runs on Python and is easy to use. PyTorch provides utilities to train and validate neural networks easily when provided with optimal architecture and network parameters. The DNNs such as Long Short term Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN), and Hybrid of TCN & LSTM and the hierarchical three stage DNN architecture are designed in PyTorch and performance metrics such as accuracy, F1 score, and runtime for inference are computed.

MATLAB [38] is used for pre-processing of ECG signal. It is a numeric computing environment which is used for a wide range of applications including signal processing. 30 minutes of ECG recordings of 48 patients is downloaded from MIT-BIH arrhythmia database and pre-processed using Pan Tompkins algorithm in MATLAB. The MATLAB simulation outputs 104,417 ECG beats and their corresponding arrhythmia type. These set of beats are given to PyTorch for classification using deep neural network (DNN). The network configuration parameters for DNN simulation in PyTorch are chosen after a thorough analysis of different combinations and examining the behavior on the testing set. Table 5.2 shows the set of finalized parameters providing best performance.

Table 5.1: Distribution of ECG Beats across each arrhythmia class

Class	No. of Beats
Normal Beat (N)	72,482
Left Bundle Branch Block Beat (L)	6,017
Right Bundle Branch Block Beat (R	7,249
Atrial Escape Beat (e)	16
Nodal (junctional) Escape Beat (j)	229
Atrial Premature Beat (A)	2,442
Aberrated Atrial Premature Beat (a)	80
Nodal (junctional) Premature Beat (J)	82
Supraventricular Premature Beat (S)	2
Fusion of Ventricular and Normal Beat (F)	793
Premature Ventricular Contraction (V)	6911
Ventricular Escape Beat (E)	104
Paced Beat (/)	7,006
Fusion of Paced and Normal Beat (f)	978
Unclassifiable Beat (Q)	26
Total	104,417

Table 5.2: Configuration Parameters

No. of hidden nodes	Stage 1	50
	LSTM	20
	BLSTM	20
	Hybrid	20
Kernel Size	TCN	4
No. of layers	LSTM	2
	BLSTM	2
	TCN	7
	Hybrid	2
Learning Algorithm	RMSProp	
Learning Rate	$2*10^{-3}$	
Epochs	50	

5.1.3. Hardware Setup

This thesis is based on the Computation in-memory (CIM) architecture proposed in ISAAC [51]. The hardware setup can be seen in Figure 2.18 for a single weight matrix. The hardware parameters considered is given in Table 5.3. Energy and area computation of the DNN accelerator is estimated based on the weight matrix dimensions. For the energy calculation, latency for one cycle considered is 100ns based on the ISAAC architecture [51].

5.2. Exploration of DNN Topologies

A comprehensive evaluation of DNN topologies was carried out to select the most suitable network topology for the hierarchical three stage DNN architecture. For the exploration of the DNN topologies, neural networks such as Long-Short-Term-Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN), and the Hybrid architecture of TCN and LSTM were considered. The DNN topology exploration results for five global arrhyth-

Table 5.3: Hardware Parameters

Component	Power (mW)	Area (mm^2)
Input Register (IR) [51]	1.24	0.0021
Output Register (OR) [51]	0.23	0.00077
Sample & Hold [51]	0.0000097	$4*10^{-8}$
Shift & Add [51]	0.05	0.00006
RRAM [51]	0.00001831	$0.152*10^{-8}$
ADC (8-bit) [51]	2	0.0012
DAC (1-bit) [51]	0.0039	$17*10^{-8}$
Activation Function [51]	0.26	0.0003
Pan Tompkins [13]	$0.55105*10^{-3}$	0.073426
PCA [57]	$144*10^{-3}$	0.06
Pointwise Multiplier [36]	0.189	0.002395

mia classes, Normal Beat (N), Supraventricular Ectopic Beat (SVEB), Ventricular Ectopic Beat (VEB), Fusion Beat (F) and Unknown Beat (Q) is given in Table 5.4 and represented graphically in Figure 5.1.

It can be seen that BLSTM provides the highest accuracy and F1 score of 98.3% and 91.1%, respectively. But BLSTM also occupies high area and energy compared to other topologies such as LSTM and Hybrid. On the other hand, LSTM has the next highest accuracy with 98.2% and F1 score of 90.2%. It consumes nearly half the energy and area than BLSTM. It also has a faster runtime execution for inference of an ECG beat. This is expected because BLSTM uses nearly double the resources compared to LSTM.

Besides, LSTM clearly outperforms TCN in all the performance metrics. The energy consumption of LSTM is more than the Hybrid topology. This is because Hybrid topology combines TCN and LSTM. However, the Hybrid topology has less accuracy and F1 score compared to LSTM. It also takes longer duration for processing ECG beats for inference and occupies more area than LSTM. Therefore, based on the accuracy, and energy trade-off, LSTM is the most efficient DNN topology selected for adaptation with the hierarchical three stage DNN architecture.

Table 5.4: Performance of DNN Topologies

	LSTM	BLSTM	TCN	Hybrid
Accuracy (%)	98.2	98.3	97.6	97.5
F1 Score (%)	90.2	91.1	88.2	88.9
Runtime (μs)	35	68.4	715	192
Energy (nJ/beat)	780.17	1523.25	1638.38	232.607
Area (mm^2)	0.0148	0.0262	0.0309	0.0182

5.3. Optimization using Dimensionality Reduction

From the previous section, it was seen that LSTM performed better compared to other DNN topologies. Therefore, several dimensionality reduction techniques were applied on LSTM for energy optimization. The results for dimensionality reduction is shown in Table 5.5 and represented graphically in Figure 5.2.

It is understandable that, smaller the feature vector of an ECG beat, lesser is the energy consumption of the network. Discrete Wavelet Transform (DWT) with 25 samples has the

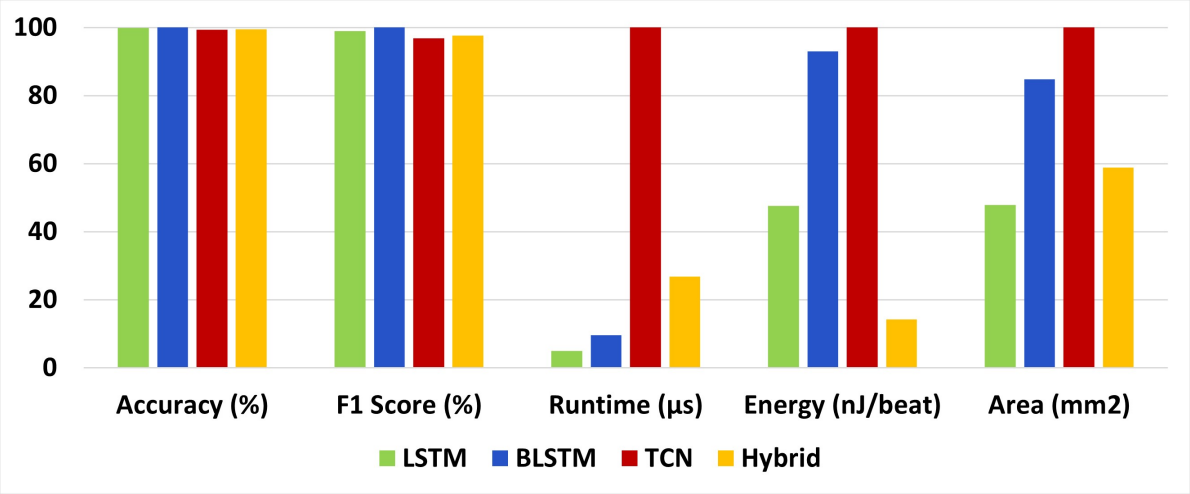

Figure 5.1: Performance comparison of Long Short term Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN), and Hybrid of TCN & LSTM. LSTM is the most optimized DNN topology.

maximum energy consumption, followed by Principal Component Analysis (PCA) with 15 samples and Linear Discriminant Analysis (LDA) with 4 samples. The accuracy and F1 score of network classification, however, depends upon how well the information of an ECG beat is represented with these dimensionality reduction techniques. As can be seen from Table 5.5, DWT and LDA show less accuracy and F1 score compared to PCA. Accuracy and F1 score of PCA is very close to the original scenario where no dimensionality reduction is applied. PCA provides 16.5x reduction in energy for LSTM topology. Therefore, based on accuracy, and energy trade-off, PCA is selected for use in the hierarchical three-stage DNN scheme.

Table 5.5: Impact of Dimensionality Reduction

	Original (250 samples)	**DWT** (25 samples)	**PCA** (15 samples)	**LDA** (5 samples)
Accuracy (%)	98.2	97.3	98.05	95.2
F1 Score (%)	90.2	85.3	90.4	79
Runtime (μs)	35	26	25	16
Energy (nJ/beat)	780.17	78.34	47.14	12.83
Area (mm^2)	0.0148	0.0148	0.0148	0.0148

5.4. Hierarchical Three Stage DNN Architecture

After the DNN exploration and dimensionality reduction, the hierarchical three stage DNN is used for offline training and inference. The results are given in Table 5.6. The confusion matrices for the three stages are given in Figure 5.3, Figure 5.4 and Figure 5.5. Stage 1, 2 and 3 obtain accuracy of 98.01%, 98.75%, and 99.46% and F1 score of 98%, 98%, and 97%, respectively.

Stage 2 and 3 use LSTM networks. It can be noticed that both Stage 2 and 3 obtain higher accuracy compared to the single stage or regular LSTM classification results obtained in Table 5.4. The regular LSTM network classifies into five output classes with fifteen subcategories obtaining an accuracy of 98.2% and F1 score of 90.2%. Stage 2 classifies the abnormal

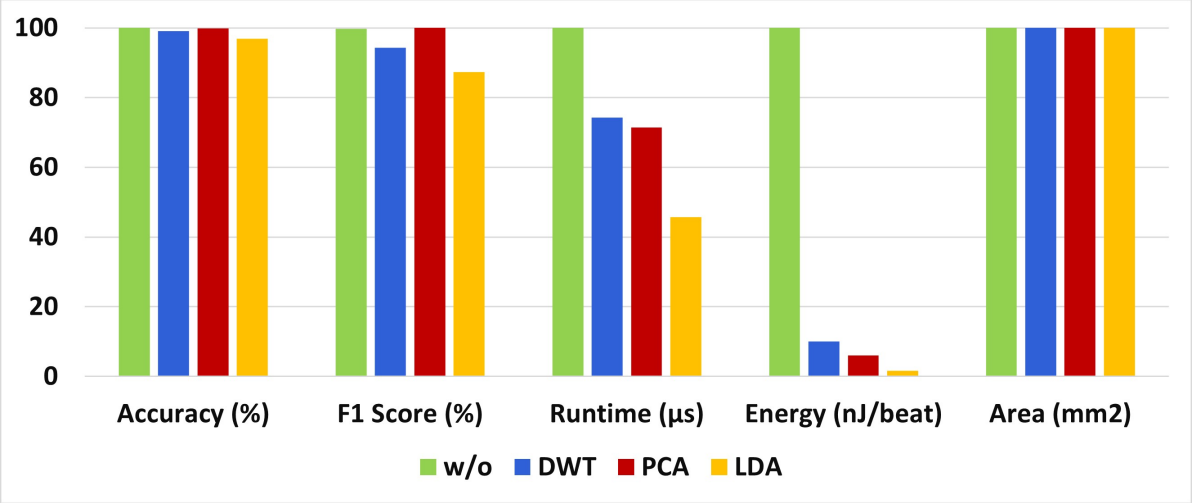

Figure 5.2: Impact of dimensionality reduction techniques. PCA is the most optimized technique maintaining high accuracy and F1 score.

beats detected by Stage 1 into four classes, having ten subcategories in total, while Stage 3 classifies into six subcategory classes. The hierarchical three stage DNN architecture obtains higher accuracy per stage because, the network is able to learn better when exposed to only a subset of ECG beats and subset of classes per stage. This results in better weight tuning and hence, the higher accuracy and F1 score of the hierarchical three stage DNN scheme.

Table 5.6: Hierarchical Three Stage DNN Performance

	Stage 1	**Stage 2**	**Stage 3**
Accuracy (%)	98.01	98.75	99.46
F1 Score (%)	98	98	97
Runtime (μs)	4	25	25
Energy (nJ/beat)	0.5647	47.39	47.41
Area (mm^2)	0.0053	0.0148	0.0148

The hierarchical three stage DNN obtains energy efficiency due to the selective network deactivation. The total energy consumption by the three stage network depends upon the number of beats processed by each stage and the energy consumption of each stage to process a single beat. Stage 1 processes all the ECG beats during inference, while Stage 2 and Stage 3 process only the true abnormal and true SVEB/VEB beats, respectively. The beats processed per stage is given in Table 5.7. The energy calculations for the hierarchical three stage DNN network is presented below:

- Energy of inference for the proposed scheme with selective network deactivation:

$$Energy_{ThreeStage} = \sum_{i=1}^{3} beats_{stage_i} * Energy_{stage_i} + Energy_{pre-processing} = 442.25\ \mu J$$

Table 5.7: Beats processed by each stage

	Stage 1	Stage 2	Stage 3
Beats	20,884	6,080	3,002

Actual \ Classified	N	Abnormal
N	14389	94
Abnormal	321	6080

Figure 5.3: Confusion Matrix for Stage 1 representing number of beats classified into Normal (N) and Abnormal classes based on 'Actual' class and 'Classified' class

Actual \ Classified	SVEB	VEB	F	Q
SVEB	173	0	0	11
VEB	0	2829	0	20
F	0	0	724	8
Q	16	15	6	2278

Figure 5.4: Confusion Matrix for Stage 2 representing number of beats classified into Supraventricular Ectopic Beat (SVEB), Ventricular Ectopic Beat (VEB), Fusion Beat (F) and Unknown Beat (Q) based on 'Actual' class and 'Classified' class

- Energy of inference without selective network deactivation (process all beats through all stages):

$$Energy_{ThreeStage} = \sum_{i=1}^{3} beats_{stage_i} * Energy_{stage_i} + Energy_{pre-processing} = 1991.59\ \mu J$$

When processing only a subset of beats through each stage by selective network deactivation, a significant amount of energy can be saved with the proposed scheme. Based on the above calculations, energy saving of up to 4.5x is achieved with the proposed scheme.

5.5. Optimization using online Partial Training

Training for the personalized system was done with online partial training of the fully connected classification layers of the hierarchical three stage DNN scheme. Improvement in accuracy and F1 score for five patients selected from the database with Patient ID's 114, 116, 118, 124, and 233 is shown in Table 5.8. Since the user's data is unique, training the network with partial training can help regain accuracy and F1 score levels as shown in Figure 5.6. Partial training is done with minimal effort and takes care of endurance problem of RRAM devices. Improvement of up to 2.02% in accuracy (Patient 118) and 2.1% in F1 score (Patient 124) is observed with online partial training.

5.6. Comparison with state-of-the-art

Table 5.9 shows the performance comparison of the proposed methodology with state-of-the-art works. Compared to the work of Hassan *et. al.* [28] and Rajpurkar *et. al.* [42], the proposed

Classified

Actual	a	A	J	S	V	E
a	34	0	0	0	0	1
A	0	694	0	0	3	3
J	0	0	3	0	1	0
S	0	0	0	1	0	0
V	0	2	0	0	2233	0
E	0	5	0	1	0	21

Figure 5.5: Stage 3 Confusion Matrix representing number of beats classified into Atrial Premature Beat (A), Aberrated Atrial Premature Beat (a), Nodal (junctional) Premature Beat (J), Supraventricular Premature Beat (S), Premature Ventricular Contraction (V), Ventricular Escape Beat (E) based on 'Actual' class and 'Classified' class

Table 5.8: Impact of partial training on Accuracy and F1 score for personalized system

Patient ID	**Accuracy**		**F1 Score**	
	Without Retraining	With Retraining	Without Retraining	With Retraining
114	98.38 %	98.64 %	94.08 %	96.08 %
116	97.67 %	98.47 %	96.54 %	97.2 %
118	95.85 %	97.87 %	96.4 %	97.11 %
124	94.06 %	96.15 %	93.2 %	95.3 %
233	96.43 %	97 %	95.5 %	96.8 %

scheme has 2.56% higher accuracy than [28] and 20.6% higher F1 score than [42]. The work by Ozal *et al.* [60] has higher accuracy and F1 score than the proposed scheme, but they have used only a subset of the dataset consisting of five arrhythmia classes. Besides, Ping *et. al.* [63] have obtained higher accuracy using all ECG arrhythmia classes but have lesser F1 score by 44.5% than the proposed scheme. Compared to the two stage neural network (TSNN) by Wang *et. al.* [56], the proposed scheme is 2.83x more energy-efficient and has 5.7% higher F1 score, with minimal area overhead of 0.01617mm^2.

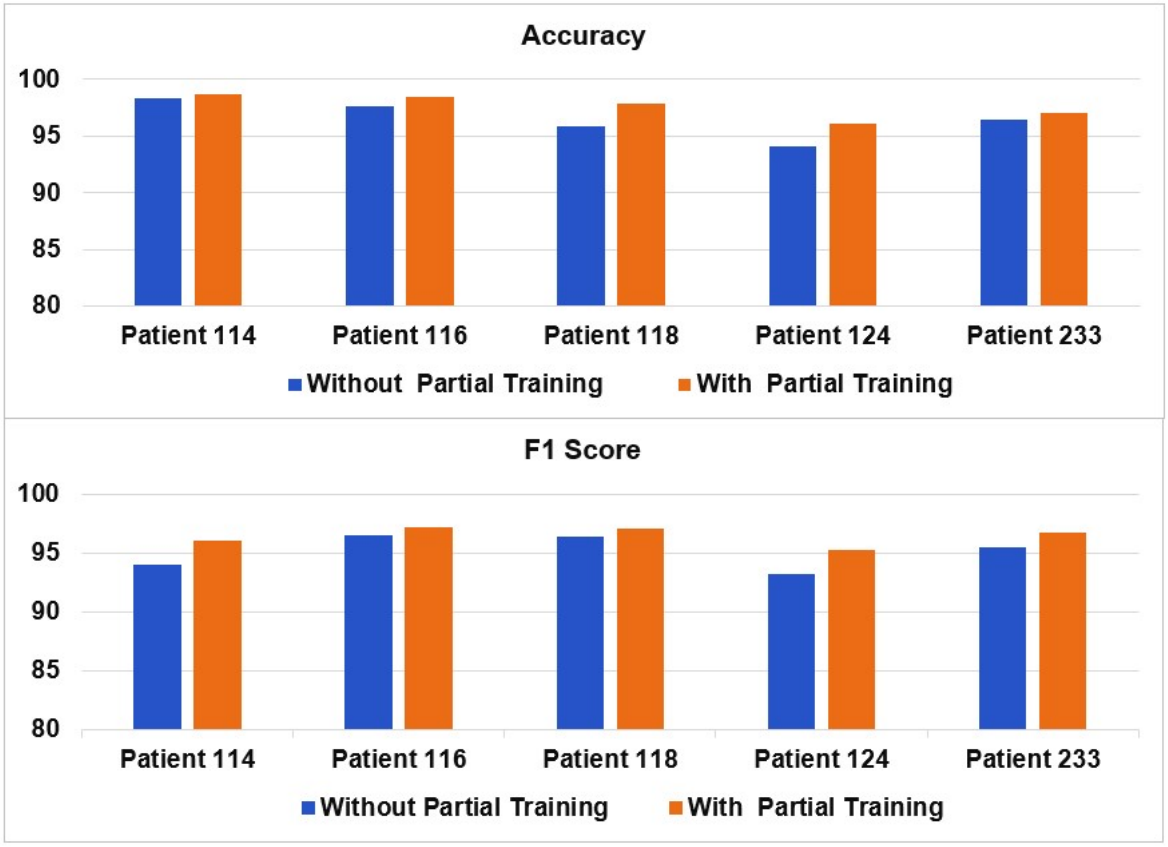

Figure 5.6: Variation of Accuracy and F1 Score for different patients with and without partial training

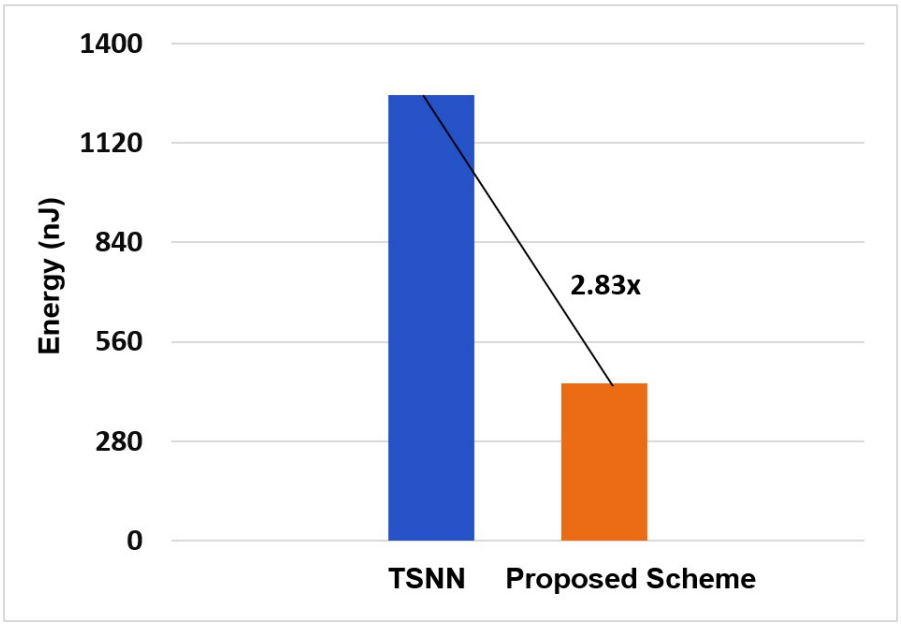

Figure 5.7: Energy improvement up to 2.83x compared to two stage neural network (TSNN) [56]

Table 5.9: Comparison with state-of-the-art

	Ozal Y., 2018 [60]	Hassan A., 2018 [28]	Rajpurkar P., 2019 [42]	Ping Z., 2020 [63]	Wang N., 2019 [56]	Proposed Scheme
No. of classes	5	5	14	15 (5 global)	15 (5 global)	15 (2 global, 7 fine)
Accuracy	99.39%	96.17%	x	99.08%	Stage 2: 98.4%	Stage1: 98% Stage2: 98.75% Stage3: 99.46% Avg = 98.73%
F1 Score	99%	x	77%	53.08%*	Stage 2: 91.9%*	Stage1: 98% Stage2: 98% Stage3: 97% Avg = 97.6%
Runtime	x	34 ms	x	x	x	54 μs
Energy	x	x	x	x	1254.864 μJ*	442.25 μJ
Area	x	x	x	x	0.07873 mm^2*	0.0949 mm^2

* Values derived from their work; hardware setup ISAAC [51]
'x' : unavailability of the information for the performance metric in the work

6

Conclusion

This chapter concludes the thesis. A summary of the thesis work is presented. Finally, potential future research related to the thesis is discussed.

6.1. Conclusion

In this thesis, a complete solution for all ECG arrhythmia classes targeting high accuracy and energy efficiency for RRAM based DNN architecture is proposed using a novel hierarchical three stage DNN solution.

Various deep neural network (DNN) topologies including Long Short Term Memory Network (LSTM), Bidirectional LSTM (BLSTM), Temporal Convolutional Network (TCN) and Hybrid of TCN and LSTM were explored to perform ECG classification and the best topology was chosen for the hierarchical three stage DNN. A comprehensive performance trade-off analysis in terms of energy and accuracy was done and LSTM was found to be the most efficient DNN topology for the proposed scheme. In order to optimize energy consumption of the DNN topology, dimensionality reduction techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) and Discrete Wavelet Transform (DWT) were applied to the chosen LSTM topology. Simulataion results showed PCA to have the best performance in terms of accuracy and energy consumption. For accuracy optimization of a personalized system, a low energy partial training approach was used to train the network online. This led to restoration of accuracy by up to 2.02% and F1 score by up to 2.1% for the new user. The hierarchical three stage DNN has a high accuracy of 98.73% and F1 score of 97.6%. The system has estimated energy consumption of 442.25μJ and occupies 0.0949mm^2 area. With the proposed scheme of selective network activation, energy-efficiency of 4.5x is obtained. Compared to state-of-the-art, the proposed scheme is 2.83x more energy-efficient. However, since more hardware resources are needed to form the three stage network, it uses higher area.

6.2. Future Works

The following suggestions are made for future research works:

- **QRS peak detection**: More advanced QRS peak detection techniques using neural network models can be adopted to provide a optimized feature extraction process.
- **Multiple lead ECG recording**: Research on using ECG signal recording from multiple leads and classifying with the neural network model will provide insight into impact of sensor fusion on network classification results.

- **Unsupervised training**: In order to generalize the system to more patients, availability of labeled dataset to train the network may be a problem. Since, the existing work uses a labeled ECG dataset for performing ECG signal classification with supervised training, alternative approaches of unsupervised training such as K-means clustering can be adopted for the system.
- **Spiking Neural Network**: Converting existing model to include spiking neurons and spike coding the ECG signal can be done to attain further energy-efficiency of the system.
- **Non-idealities of the RRAM devices**: One of the critical challenges while designing RRAM based CIM architectures are the device non-idealities. Impact of device non-idealities on accuracy of the system can be evaluated in future works. Periodic device training to program the RRAM devices and the cost of the approach can be estimated. Besides, since online personalized training is carried out, endurance problems of the RRAM devices can be analyzed.

A

Appendix

Pan Tompkins

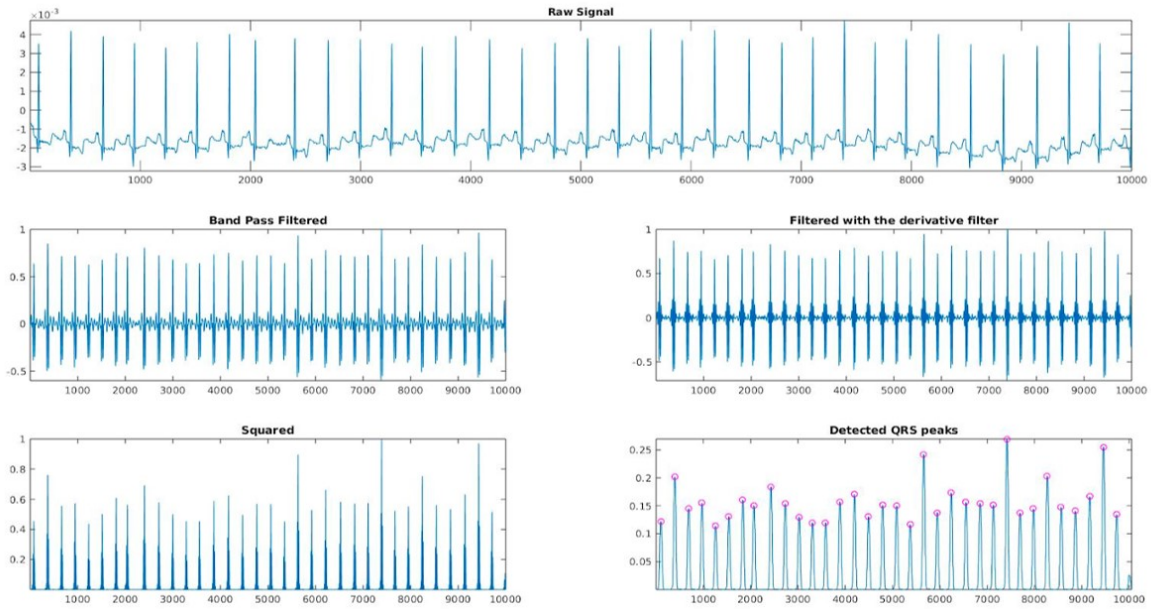

Figure A.1: MATLAB Simulation of QRS Peak Detection using Pan Tompkins algorithm for Patient ID 100

LSTM

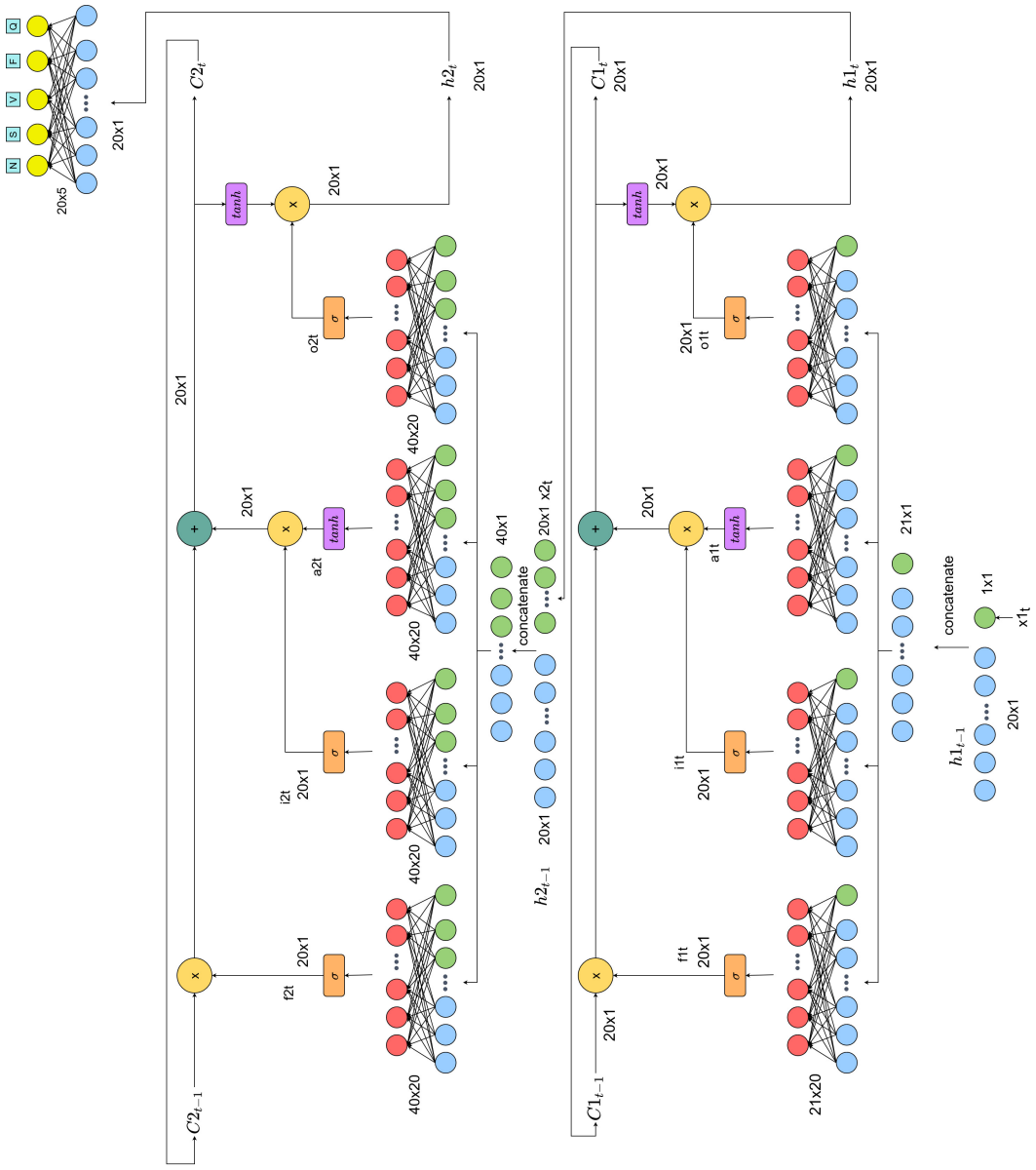

Figure A.2: A detailed representation of inside 2 layer LSTM Network

Let no. of samples in ECG beat = m = 250
Total Energy = m*(1.6597mW*100ns + 3.4032mW*100ns + 26.13mW*100ns) + (1.332mW*100ns + 1.7016mW*100ns + 0.54mW*100ns)
= 780.17 nJ

BLSTM

No. of samples in ECG beat, m = 250
Total Energy = m*(2.079mW*100ns + 6.8064mW*100ns + 52.03mW*100ns) + (1.4256mW*100ns + 2mW*100ns + 0.54*100ns)
= 250*60.9154 + 0.39656 nJ
= 1523.25 nJ

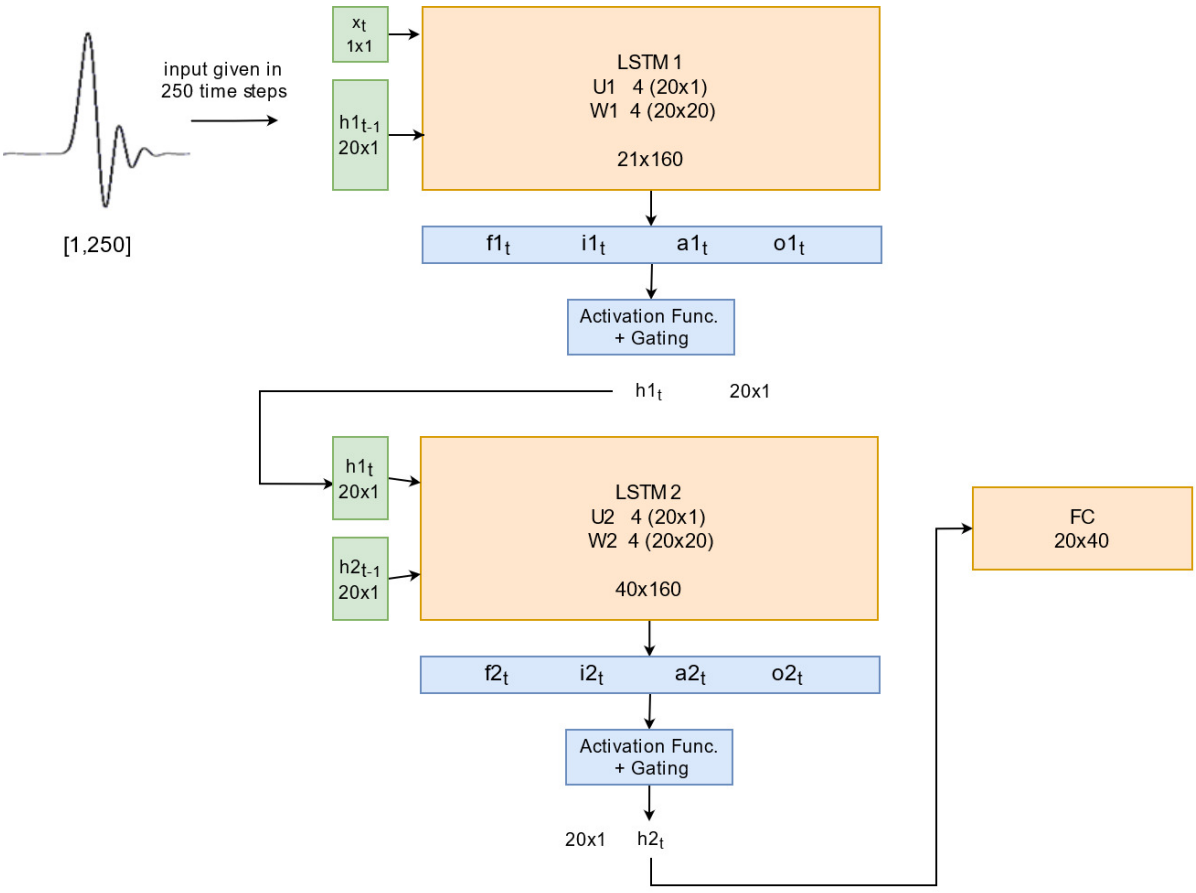

Figure A.3: A 2 layer LSTM Network Structure

Table A.1: LSTM Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	2 ADC (7-bit)	3.4032	OR	0.23
61 DAC	0.2379			2x60 PW mult.	22.68
21x160 U/W layer 1 crossbar	0.06152			12 act. function	3.12
40x160 U/W layer 2 crossbar	0.11718			2 S&A	0.1
320 S&H	0.003104				
Total	1.6597 mW	3.4032 mW		26.13 mW	

Table A.2: FC LSTM Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	1 ADC (7-bit)	1.7016	OR	0.23
20 DAC	0.078			1 act. function	0.26
20x40 crossbar	0.0146			1 S&A	0.05
40 S&H	0.000388				
Total	1.332 mW	1.7016 mW		0.54 mW	

Table A.3: LSTM Area Calculation

Component	Area (mm2)
IR	0.0021
81 DAC	$13.77*10^{-6}$
21x160 U/W layer 1 crossbar	$5.1*10^{-6}$
40x160 U/W layer 2 crossbar	$9.72*10^{-6}$
20x40 FC crossbar	$1.21*10^{-6}$
360 S&H	$14.4*10^{-6}$
3 ADC (7-bit)	0.00315
OR	0.00077
2 PW mult.	0.00479
13 act. function	0.0039
3 S&A	0.00018
Total	0.01489 mm2

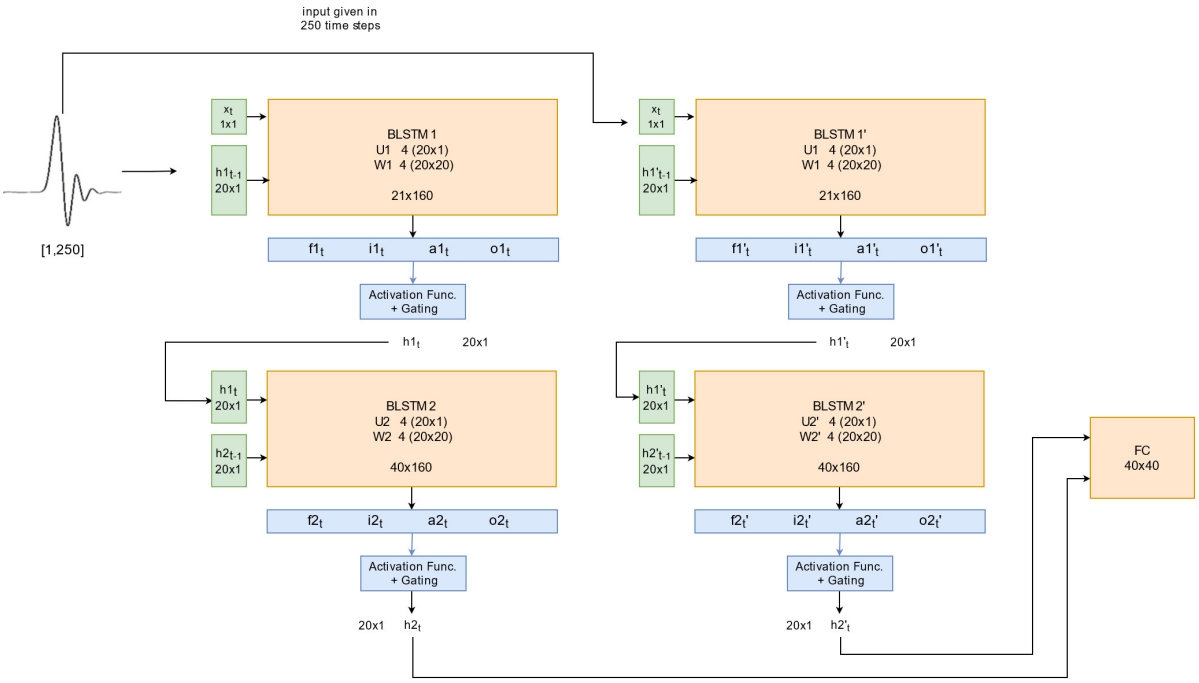

Figure A.4: A 2 layer BLSTM Network Structure

Table A.4: BLSTM Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	4 ADC (7-bit)	6.8064	OR	0.23
2*61 DAC	0.4758			2x2x60 PW mult.	45.36
2 21x160 U/W crossbar	0.123			24 act. function	6.24
2 40x160 U/W2 crossbar	0.23436			4 S&A	0.2
2*320 S&H	0.006208				
Total	2.079 mW	6.8064 mW		26.13 mW	

Table A.5: FC BLSTM Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	1 ADC (8-bit)	2	OR	0.23
40 DAC	0.156			1 act. function	0.26
40x40 crossbar	0.029296			1 S&A	0.05
40 S&H	0.000388				
Total	1.4256 mW	2 mW		0.54 mW	

Table A.6: BLSTM Area Calculation

Component	Area (mm2)
IR	0.0021
162 DAC	$27.54*10^{-6}$
2 21x160 U/W crossbar	$10.2*10^{-6}$
2 40x160 U/W layer 2 crossbar	$19.2*10^{-6}$
40x40 FC crossbar	$1.21*10^{-6}$
680 S&H	$27.2*10^{-6}$
5 ADC (4 7-bit, 1 8-bit)	0.006
OR	0.00077
4 PW mult.	0.00958
25 act. Function	0.0075
5 S&A	0.0003
Total	0.02625mm2

TCN

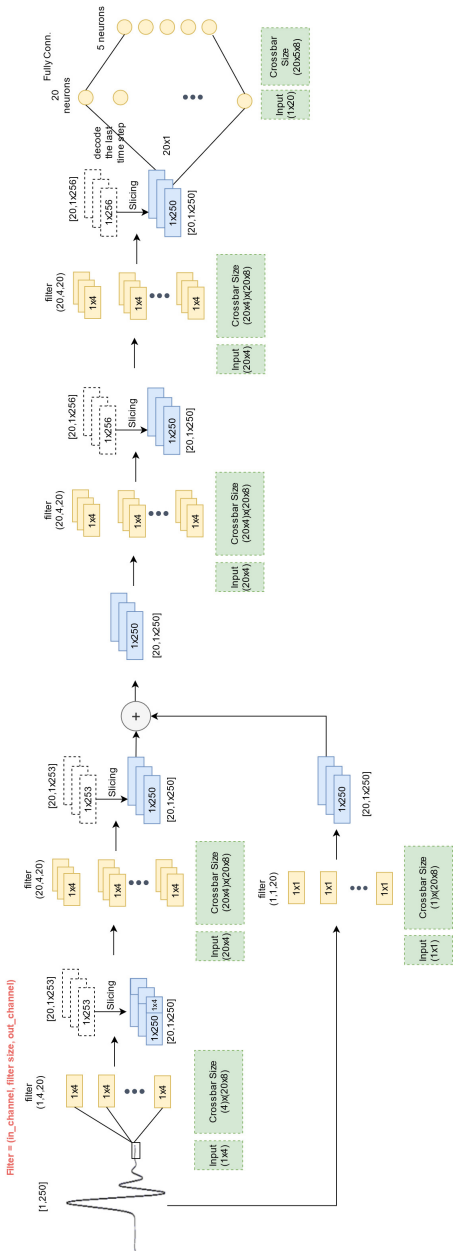

Figure A.5: A 2 layer TCN structure

No. of convolution operations in layer i ECG beat = mi
Total Energy = $\sum_{i=1}^{L=7}(mi * Power_{Layrer_i}$*100ns) + PowerFC *100ns
= 253*6.714mW*100ns + (256*7.8177mW*100ns + 262*7.8177mW*100ns + 274*7.8177mW*100ns + 298*7.8177mW*100ns + 346*7.8177mW*100ns + 442*7.8177mW*100ns) + (1.332mW*100ns + 1.7016mW*100ns + 0.54mW*100ns)
= 1638.38 nJ

Table A.7: TCN Energy Consumption, Layer 1

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	3 ADC (2,4,9-bit)	3.729	OR	0.23
85 DAC	0.3315			3 act. function	0.78
4x160 crossbar	0.0117			3 S&A	0.15
80x160 crossbar	0.2343				
1x160 crossbar	0.0029				
3*160 S&H	0.00465				
Total	1.825 mW	3.729 mW		1.16 mW	

Table A.8: TCN Energy Consumption, Layer >=2

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	2 ADC (9-bit)	4.632	OR	0.23
160 DAC	0.624			2 act. function	0.52
2 80x160 crossbar	0.4686			2 S&A	0.1
2*160 S&H	0.0031				
Total	2.3357 mW	4.632 mW		0.85 mW	

Table A.9: FC TCN Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	1 ADC (7-bit)	1.7016	OR	0.23
20 DAC	0.078			1 act. function	0.26
20x40 crossbar	0.0146			1 S&A	0.05
40 S&H	0.000388				
Total	1.332 mW	1.7016 mW		0.54 mW	

Table A.10: TCN Area Calculation

Component	Area (mm2)
IR	0.0021
105 DAC + 7*160 DAC	$208.25*10^{-6}$
4x160 crossbar	$0.972*10^{-6}$
80x160 + 7*2 80x160 crossbar	$291.81*10^{-6}$
1x160 crossbar	$0.243*10^{-6}$
20x40 FC crossbar	$1.21*10^{-6}$
520 S&H + 7*2*160 S&H	$110.4*10^{-6}$
4 S&A + 7*2 S&A	0.00108
OR	0.00077
ADC (2, 4,7, 9-bit) + 7*2 ADC (9-bit)	0.018
4 act. Function + 7*2 act. Function	0.009
Total	0.03095 mm2

Hybrid TCN+LSTM

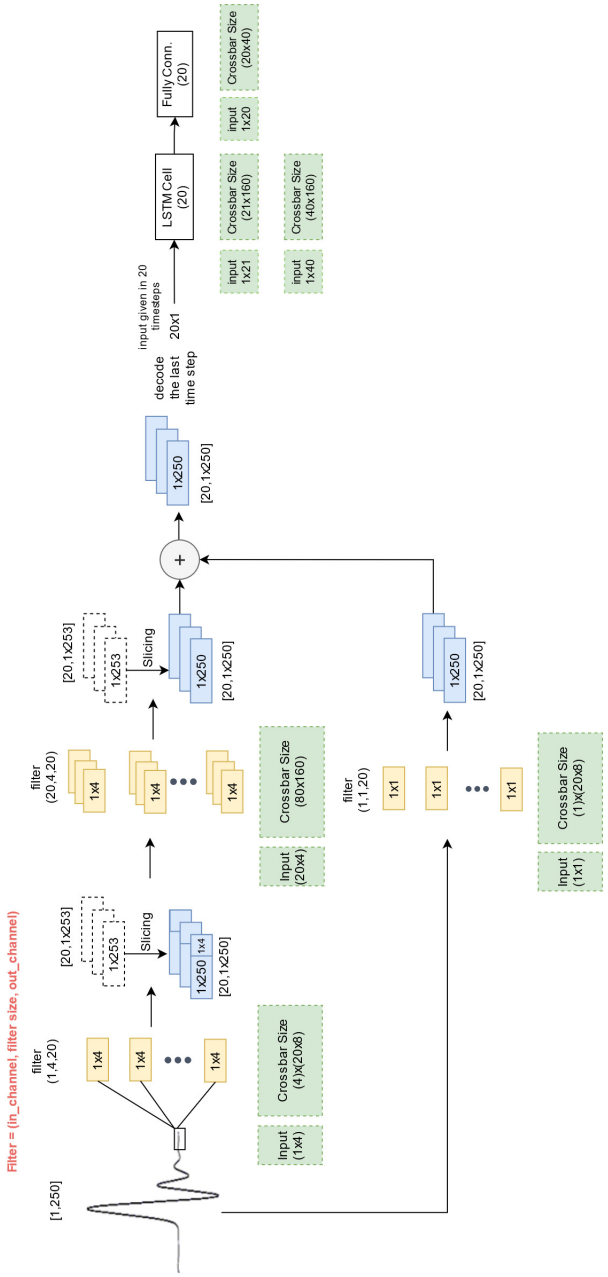

Figure A.6: Hybrid Network Structure

Total ECG beat samples= 250, No. of convolution operations in layer 1= mi = 253, LSTM hidden nodes, n=20
Total Energy = $\sum_{i=1}^{L=7}(mi * Power_{Layrer_i}$*100ns) + PowerFC *100ns
= 253*(1.825mW*100ns+3.729mW*100ns + 1.16mW*100ns) + 20*(1.6597mW*100ns + 3.4032mW*100ns + 26.13mW*100ns) +
(1.332mW*100ns + 1.7016mW*100ns + 0.54mW*100ns)
= 232.607 nJ

Table A.11: Hybrid Layer 1 Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	3 ADC (2,4,9-bit)	3.729	OR	0.23
85 DAC	0.3315			3 act. function	0.78
4x160 crossbar	0.0117			3 S&A	0.15
80x160 crossbar	0.2343				
1x160 crossbar	0.0029				
3*160 S&H	0.00465				
Total	1.825 mW	3.729 mW		1.16 mW	

Table A.12: Hybrid layer 2 Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	2 ADC (7-bit)	3.4032	OR	0.23
61 DAC	0.2379			2x60 PW mult.	22.68
21x160 U/W layer 1 crossbar	0.06152			12 act. function	3.12
40x160 U/W layer 2 crossbar	0.11718			2 S&A	0.1
320 S&H	0.003104				
Total	1.6597 mW	3.4032 mW		26.13 mW	

Table A.13: FC Hybrid Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	1 ADC (7-bit)	1.7016	OR	0.23
20 DAC	0.078			1 act. function	0.26
20x40 crossbar	0.0146			1 S&A	0.05
40 S&H	0.000388				
Total	1.332 mW	1.7016 mW		0.54 mW	

Table A.14: Hybrid architecture Area calculation

Component	Area (mm2)
IR	0.0021
166 DAC	$28.22*10^{-6}$
4x160 crossbar	$0.972*10^{-6}$
80x160 crossbar	$19.44*10^{-6}$
1x160 crossbar	$0.243*10^{-6}$
21x160 U/W layer 1 crossbar	$5.1*10^{-6}$
40x160 U/W layer 2 crossbar	$9.72*10^{-6}$
20x40 FC crossbar	$1.21*10^{-6}$
840 S&H	$33.6*10^{-6}$
6 ADC (2, 4, 3 7-bit, 9)	0.0054
OR	0.00077
16 act. Function	0.0048
6 S&A	0.00036
2 PW mult.	0.00479
Total	0.01822 mm2

Stage 1

Figure A.7: Stage 1 crossbar array

Table A.15: Stage 1 Energy Consumption

0-100ns		**100-200ns**		**200-300ns**	
IR	1.24	1 6-bit + 1 8-bit ADC	3.4354	OR	0.23
15 + 50 = 65 DAC	0.2535			1 act. function	0.26
1 15x(50x8) FC crossbar	0.1098			2 S&A	0.1
1 50x(2x8) FC crossbar	0.0146				
50*8 + 2*8 = 416 S&H	0.00403				
Total	1.62193 mW	3.4354 mW		0.59 mW	

Table A.16: Stage 1 Area Calculation

Component	Area (mm2)
IR	0.0021
15 + 50 = 65 DAC	$0.1105*10^{-6}$
1 15x(50x8) FC crossbar	$0.0912*10^{-6}$
1 50x(2x8) FC crossbar	$1.216*10^{-6}$
50*8 + 2*8 = 416 S&H	$16.64*10^{-6}$
1 6-bit + 1 8-bit ADC	0.0021
OR	0.00077
1 ReLU act. function	0.0003
2 S&A	0.00012
Total	0.00539mm2

Stage 1 Energy = (1.62193mW*100ns + 3.4354mW*100ns + 0.59mW*100ns) = 0.56473 nJ

Bibliography

[1] https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[2] https://computingcell.com/memristor-technology-g etting-closer/.

[3] https://en.wikipedia.org/wiki/discrete_wavelet_transform.

[4] https://en.wikipedia.org/wiki/linear_discriminant_analysis.

[5] https://en.wikipedia.org/wiki/principal_component_analysis.

[6] https://towardsdatascience.com/linear-discriminant-analysis-in-python-76b8b17817c2.

[7] https://www.world-heart-federation.org/resources/cardiovascular-disease-infographic/.

[8] Paszke Adam et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[9] Alireza Amirshahi and Matin Hashemi. Ecg classification algorithm based on stdp and r-stdp neural networks for real-time monitoring on ultra low-power personal wearable devices. *IEEE Transactions on Biomedical Circuits and Systems (TBioCAS)*, 13(6):1483–1493, December 2019.

[10] Adil Arish, Hassan Abid, Nadir Najib, Usama Jillani, Wala Saadeh, and Muhammad Awais Bin Altaf. Wearable ecg measurement system for detection of cardiac arrhythmia. In *2019 UK/ China Emerging Technologies (UCET)*, pages 1–4, 2019.

[11] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018.

[12] Felix Christian Bauer, Dylan Richard Muir, and Giacomo Indiveri. Real-time ultra-low power ecg anomaly detection using an event-driven neuromorphic processor. *IEEE Transactions on Biomedical Circuits and Systems*, 13(6):1575–1582, Dec 2019.

[13] Nourhan Bayasi, Hani Saleh, Baker Mohammad, and Mohammed Ismail. 65-nm asic implementation of qrs detector based on pan and tompkins algorithm. pages 84–87, 11 2014.

[14] S. Chauhan and L. Vig. Anomaly detection in ecg time signals via deep long short-term memory networks. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–7, 2015.

[15] L. Chua. Memristor-the missing circuit element. *IEEE Transactions on Circuit Theory*, 18(5):507–519, 1971.

[16] F. Corradi, S. Pande, J. Stuijt, N. Qiao, S. Schaafsma, G. Indiveri, and F. Catthoor. Ecg-based heartbeat classification in neuromorphic hardware. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2019.

[17] Javier del Valle, Juan Ramirez, Marcelo Rozenberg, and Ivan Schuller. Challenges in materials and devices for resistive-switching-based neuromorphic computing. *Journal of Applied Physics*, 124:211101, 12 2018.

[18] Lei Deng et al. Rethinking the performance comparison between snns and anns. *Neural Networks 121*, pages 294 – 307, 2020.

[19] Zümray Dokur and Tamer Ölmez. Ecg beat classification by a novel hybrid neural network. *Computer Methods and Programs in Biomedicine*, 66(2):167–181, 2001.

[20] Xunde Dong. Ecg beat classification via deterministic learning. *Neurocomputing*, 240:1–12, 2017.

[21] R. Acharya et al. Classification of cardiac abnormalities using heart rate signals. *Medical and Biological Eng. and Computing*, pages 288–293, 2004.

[22] R. Lozano et al. Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the global burden of disease study 2010. *The Lancet, vol. 380, no. 9859, pp. 2095*, 2013.

[23] Alibart F et al. Pattern classification by memristive crossbar circuits using ex situ and in situ training. In *Nat Commun 4, 2072 (2013)*, 2013.

[24] Zahoor F et al. Resistive random access memory (rram): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications. *Nanoscale Res Lett 15, 90 (2020)*, 1974.

[25] Zahoor F. et al. Resistive random access memory (rram): an overview of materials, switching mechanism, performance, multilevel cell (mlc) storage, modeling, and applications. *Nanoscale Res Lett 15, 90 (2020).*, 2020.

[26] W.H. Freeman. Environmental science for ap. 2012.

[27] İnan Güler and Elif Derya Übeylı˙. Ecg beat classifier designed by combined neural network model. *Pattern Recognition*, 38(2):199–208, 2005.

[28] A. M. Hassan. Real-time cardiac arrhythmia classification using memristor neuromorphic computing system. In *2018 40th Annual International Conference of the IEEE EMBC*, 2018.

[29] John L Hennessy and David A Patterson. Computer architecture: a quantitative approach. *Elsevier,*, 2011.

[30] Sepp Hochreiter et al. Long short term memory network. *Neural Computation*, 1997.

[31] S. Jain, A. Ankit, I. Chakraborty, T. Gokmen, M. Rasch, W. Haensch, K. Roy, and A. Raghunathan. Neural network accelerator design with resistive crossbars: Opportunities and challenges. *IBM Journal of Research and Development*, 63(6):10:1–10:13, 2019.

[32] Chankyu Lee, Syed Sarwar, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Enabling spike-based backpropagation for training deep neural network architectures. *Frontiers in Neuroscience*, 14:119, 02 2020.

[33] J. Lee, K. L. Park, M. H. Song, and K. J. Lee. Arrhythmia classification with reduced features by linear discriminant analysis. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 1142–1144, 2005.

[34] M. Li, T.; Zhou. Ecg classification using wavelet packet entropy and random forests. *Entropy*, 2016.

[35] Y. Lin, C. Chuang, C. Yen, S. Huang, P. Huang, J. Chen, and S. Lee. Artificial intelligence of things wearable system for cardiac disease detection. *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 67–70, 2019.

[36] V Manu, A M Vijaya Prakash, and Mohan U Chandra. Design and implementation of sixteen-bit low power and area efficient dadda multiplier. In *2019 4th International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT)*, pages 631–636, 2019.

[37] Ingolfsson Thorir Mar et al. Ecg-tcn: Wearable cardiac arrhythmia detection with a temporal convolutional network. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 1–4, 2021.

[38] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[39] Baker Mohammad, Maguy Abi Jaoude, Vikas Kumar, Dirar Mohammad Al Homouz, Heba Abu Nahla, Mahmoud Al-Qutayri, and Nicolas Christoforou. State of the art of metal oxide memristor devices. *Nanotechnology Reviews*, 5(3):311–329, 2016.

[40] Mark RG. Moody GB. The impact of the mit-bih arrhythmia database. *IEEE Eng in Med and Biol*, pages 20(3):45–50, (May-June 2001).

[41] Faust O et al. Automated detection of atrial fibrillation using long short-term memory network with rr interval signals. *Comput Biol Med*, Nov 2018.

[42] Rajpurkar P et al. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat Med 25*, 2019.

[43] Willis J. Pan, Jiapu; Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering.*, page 230–236., (March 1985).

[44] Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Front Neuroscience*, 2018.

[45] Amit Prakash, Debanjan Jana, and Siddheswar Maikap. Taox-based resistive switching memories: Prospective and challenges. *Nanoscale research letters*, 8:418, 10 2013.

[46] Xia Q. and Yang J.J. Memristive crossbar arrays for brain-inspired computing. *Nat. Mater. 18*, page 309–323, 2019.

[47] F. Qiao, B. Li, Y. Zhang, H. Guo, W. Li, and S. Zhou. A fast and accurate recognition of ecg signals based on elm-lrf and blstm algorithm. *IEEE Access*, 8:71189–71198, 2020.

[48] S. Raj and K. C. Ray. Ecg signal analysis using dct-based dost and pso optimized svm. *IEEE Transactions on Instrumentation and Measurement*, 66(3):470–478, 2017.

[49] Choi S et al. Data clustering using memristor networks. In *Sci Rep 5, 10492*, 2015.

[50] A. T. Sadiq and N. H. Shukr. Classification of cardiac arrhythmia using id3 classifier based on wavelet transform. *Iraqi J. of Sci.*, pages 1167–1175, 2013.

[51] Ali Shafiee et al. Isaac: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pages 14–26, 2016.

[52] Dmitri B. Strukov et al. The missing memristor found. *Nature, 453(7191)*, 2008.

[53] Jen Hong Tana et al. Application of stacked convolutional and long short-term memory networkfor accurate identification of cad ecg signals. *Computers in Biology and Medicine*, 2018.

[54] M. S. Tarkov. Mapping neural network computations onto memristor crossbar. In *2015 International Siberian Conference on Control and Communications (SIBCON)*, 2015.

[55] Sudeshna Dash Thilagavathy R. Real-time ecg signal feature extraction and classification using support vector machine. In *2020 International Conference on Contemporary Computing and Applications (IC3A)*, pages 44–48, 2020.

[56] N. Wang, J. Zhou, G. Dai, J. Huang, and Y. Xie. Energy-efficient intelligent ecg monitoring for wearable devices. *IEEE Transactions on Biomedical Circuits and Systems*, 13(5):1112–1121, 2019.

[57] Tong Wu, Wenfeng Zhao, Hongsun Guo, Hubert Lim, and Zhi Yang. A streaming pca based vlsi chip for neural data compression. In *2016 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 192–195, 2016.

[58] Bonan Yan et al. Resistive memory-based in-memory computing: From device and large-scale integration system perspectives. *Advanced Intelligent Systems*, 2019.

[59] He Yangdong et al. Temporal convolutional networks for anomaly detection in time series. *Journal of Physics: Conference Series*, 2019.

[60] Ozal Yildirim. A novel wavelet sequence based on deep bidirectional lstm network model for ecg signal classification. *Computers in Biology and Medicine*, 96, 03 2018.

[61] Ozal Yildirim et al. Arrhythmia detection using deep convolutional neural network with long duration ecg signals. *Computers in Biology and Medicine*, 102:411–420, 2018.

[62] Ozal Yildirim et al. A new approach for arrhythmia classification using deep coded features and lstm networks. *Computer Methods and Programs in Biomedicine*, 2019.

[63] Ping Zhang et al. Classification of ecg signals based on lstm and cnn. *International Conference on Artificial Intelligence and Security*, 2020.