# Severity-based Hierarchical ECG Classification using Neural Networks

Diware, S.S.; Dash, Sudeshna ; Gebregiorgis, A.B.; Joshi, Rajiv V.; Strydis, C.; Hamdioui, S.; Bishnoi, R.K.

**DOI**
10.1109/TBCAS.2023.3242683

**Publication date**
2023

**Document Version**
Final published version

**Published in**
IEEE Transactions on Biomedical Circuits and Systems

# Severity-Based Hierarchical ECG Classification Using Neural Networks

Sumit Diware, Sudeshna Dash, Anteneh Gebregiorgis, *Member, IEEE*, Rajiv V. Joshi, *Fellow, IEEE*, Christos Strydis, *Senior Member, IEEE*, Said Hamdioui, *Senior Member, IEEE*, and Rajendra Bishnoi

***Abstract*—Timely detection of cardiac arrhythmia characterized by abnormal heartbeats can help in the early diagnosis and treatment of cardiovascular diseases. Wearable healthcare devices typically use neural networks to provide the most convenient way of continuously monitoring heart activity for arrhythmia detection. However, it is challenging to achieve high accuracy and energy efficiency in these smart wearable healthcare devices. In this work, we provide architecture-level solutions to deploy neural networks for cardiac arrhythmia classification. We have created a hierarchical structure after analyzing various neural network topologies where only required network components are activated to improve energy efficiency while maintaining high accuracy. In our proposed architecture, we introduce a severity-based classification approach to directly help the users of the wearable healthcare device as well as the medical professionals. Additionally, we have employed computation-in-memory based hardware to improve energy efficiency and area consumption by leveraging in-situ data processing and scalability of emerging memory technologies such as resistive random access memory (RRAM). Simulation experiments conducted using the MIT-BIH arrhythmia dataset show that the proposed architecture provides high accuracy while consuming average energy of 0.11 $\mu$J per heartbeat classification and 0.11 mm$^2$ area, thereby achieving 25$\times$ improvement in average energy consumption and 12$\times$ improvement in area compared to the state-of-the-art.**

***Index Terms*—Arrhythmia, computation-in-memory, ECG, neural networks, resistive random access memory (RRAM), severity-based classification.**

## I. Introduction

THE heart plays an important role in human survival and any heart-related disorders, commonly known as *cardiovascular diseases* (CVDs), can present a significant danger to human life. CVDs are reported to be one of the leading causes of death worldwide [1] and are estimated to cause up to 23 million deaths by 2030 [2]. Diagnosis of CVDs at an early stage can facilitate timely medical treatment and greatly reduce CVD-related health risks. Such early diagnosis can be achieved by detecting the abnormal activity of the heart known as *arrhythmia*. There exist several types of arrhythmia based on the manner in which the heart activity deviates from its normal behavior. Timely detection of various types of arrhythmia requires monitoring of the activity of the heart. Wearable healthcare devices provide the most convenient way of achieving such monitoring. These devices are equipped with sensors that can record the heart activity in the form of *electrocardiogram* (ECG) signal. The task of identifying various types of arrhythmia is then expressed as the classification of heartbeats in the recorded ECG signal into different arrhythmia types (classes). As neural networks are inherently best suited for such classification tasks, these devices use neural networks as ECG classifiers to automatically identify various types of arrhythmia.

The neural network-based ECG classifier in a wearable healthcare device should have high classification accuracy to detect various arrhythmia types correctly. Moreover, it has to be energy-efficient as wearable healthcare devices are battery-powered and thereby have limited energy resources. Its classification outputs should also indicate the severity impact of detected arrhythmia classes which can help the users in knowing how urgently they need to seek medical attention, which can potentially prove to be life-saving. However, state-of-the-art neural network-based ECG classifiers fail to meet these requirements. Many works have adopted neural networks with a large number of layers to obtain high accuracy [3], [4], [5]. This results in high energy consumption as such big neural networks require a lot of hardware resources. Most of the existing works just focus on developing ECG classification models without taking into account the implications on hardware performance metrics such as energy [6], [7], [8], [9], [10], [11], [12], [13], [14]. Moreover, none of them take the severity impact into account. Hence, there is a strong need for ECG classification hardware that can deliver high accuracy and energy efficiency while also considering severity impact.

In this paper, we address the challenge of designing a severity-based, accurate, and energy-efficient ECG classifier. We first create a classification architecture that consists of multiple small sub-classifiers connected in a hierarchical manner instead of a single large and complex classifier. Each sub-classifier deals

Manuscript received 30 September 2022; revised 2 December 2022 and 16 January 2023; accepted 2 February 2023. Date of publication 6 February 2023; date of current version 23 March 2023. This work was supported by the ECSEL Joint Undertaking (JU) through the EU H2020 DAIS under Grant Agreement 101007273. This paper was recommended by Associate Editor Benoit Gosselin. *(Corresponding author: Sumit Diware.)*

Sumit Diware, Anteneh Gebregiorgis, Said Hamdioui, and Rajendra Bishnoi are with the Computer Engineering Department, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: S.S.Diware@tudelft.nl; a.b.gebregiorgis@tudelft.nl; s.hamdioui@tudelft.nl; r.k.bishnoi@tudelft.nl).

Sudeshna Dash is with the ASML Holding N.V., 5504 DR Veldhoven, The Netherlands (e-mail: sudeshna.dash@asml.com).

Rajiv V. Joshi is with the IBM Thomas J. Watson Research Centre, Yorktown Heights, NY 10598 USA (e-mail: rvjoshi@us.ibm.com).

Christos Strydis is with the Erasmus Medical Center, Delft University of Technology, 3015 CN Rotterdam, The Netherlands (e-mail: c.strydis@erasmusmc.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TBCAS.2023.3242683.

Digital Object Identifier 10.1109/TBCAS.2023.3242683

with only a subset of arrhythmia classes which leads to good accuracy. This hierarchical design also allows us to activate various sub-classifiers only when needed, thereby saving energy. The proposed architecture uses a novel severity-based activation structure for sub-classifiers. The top levels of the hierarchy indicate how quickly the user should seek medical attention. The bottom hierarchical levels help the doctors in diagnosis (the process of finding the physiological root cause of arrhythmia) and then prescribing treatment (medicines, medical procedures, etc.) for arrhythmia based on the diagnosis. Moreover, we propose a hardware design methodology for each internal sub-classifier using the most energy-efficient neural network while still ensuring good accuracy. This hardware design is based on the computation-in-memory (CIM) paradigm which uses emerging memory technologies such as resistive random access memory (RRAM) to provide higher energy efficiency compared to conventional von-Neumann architecture-based implementation for neural networks.

The key contributions of this paper are:

- We develop a hierarchical classification architecture that breaks down the full classification task into smaller subtasks to achieve high accuracy and activates various architectural components only when required in order to save energy.
- We propose a severity-based activation structure that helps the users in seeking timely medical attention as well as helps the medical professional in speeding up the diagnosis and treatment.
- We provide a methodology for the hardware design of various components in the hierarchical ECG classification architecture using RRAM-based computation-in-memory paradigm to achieve the best balance between energy efficiency and accuracy.

Simulation results show that the proposed architecture consumes an average energy of 0.11 $\mu$J per heartbeat classification and requires 0.11 mm$^2$ area, which results in 25$\times$ less average energy consumption and 12$\times$ less area compared to the state-of-the-art while maintaining high accuracy.

The rest of the paper is organized as follows: Section II describes the basics of ECG and neural network-based arrhythmia classification. Sections III, IV and V provide the design and implementation details of the proposed severity-based energy-efficient ECG classifier design. This is followed by simulation setup details in Section VI and simulation results in Section VII. Finally, Section VIII concludes the paper.

## II. Background

### A. Electrocardiogram (ECG)

The human heart is made up of four chambers. The upper two chambers are called *atria* and the lower two chambers are called *ventricles*. These chambers undergo contraction and relaxation in a periodic manner. This activity can be recorded as a graph of voltage versus time known as *electrocardiogram* (ECG). A single ECG recording contains multiple cycles of contraction and relaxation of the heart chambers. These cycles are known as ECG beats. A visualization of an ECG beat is shown in Fig. 1

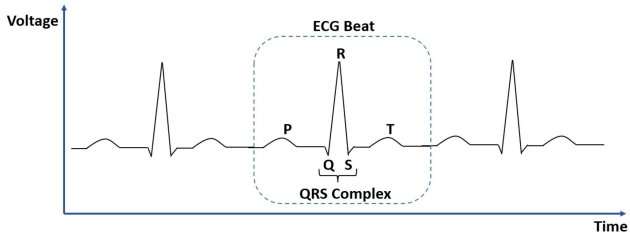



Fig. 1. ECG signal illustration with 'PQRST' cycle for an ECG beat, where P: atrial contraction, Q: interventricular septum contraction, R: ventricular contraction (main mass), S: ventricular contraction (at heart's base), and T: ventricular relaxation. Atrial relaxation is obscured by QRS complex.

TABLE I
AAMI [17] Grouping of ECG Arrhythmia Classes in MIT-BIH Dataset [15]

| AAMI Class | Arrhythmia Class |
|---|---|
| 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) |
| Supraventricular Ectopic Beat (S) | 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) |
| Ventricular Ectopic Beat (V) | Premature Ventricular Contraction (V) |
| | Ventricular Escape Beat (E) |
| Unknown Beat (Q) | Paced Beat (/) |
| | Fusion of Paced and Normal Beat (f) |
| | Unclassifiable Beat (Q) |

which begins with the contraction of atria represented by 'P'. This is followed by relaxation of the atria and contraction of the ventricles observed as the 'QRS' complex. 'Q' wave represents the contraction of the interventricular septum. 'R' wave indicates the contraction of the main mass of the ventricles. 'S' wave denotes the contraction of the ventricles at the base of the heart. The beat ends when the ventricles undergo relaxation denoted as 'T'. When a recorded ECG beat deviates from its expected normal behavior, it represents the abnormal activity of the heart chambers called *arrhythmia*. There exist several different classes (types) of arrhythmia based on the exact manner in which the recorded ECG beat deviates from its normal behavior. For instance, MIT-BIH Arrhythmia dataset [15] (provided through PhysioNet [16]) consists of 15 arrhythmia classes which are further grouped into 5 superclasses by *Association for the Advancement of Medical Instrumentation* (AAMI) [17] as shown in Table I. The arrhythmia classes can be distinguished from each other (as well as the normal heart activity) by using different features of the 'QRS' complex such as timing, amplitude, etc. Hence, the 'QRS' complex in an ECG beat plays a crucial role in identifying arrhythmia classes.

### B. Arrhythmia Detection

Activity of the heart should be regularly monitored for timely detection of arrhythmia. This involves recording the ECG signal and identifying the types of abnormal beats in it. Various approaches used for such monitoring are as follows:

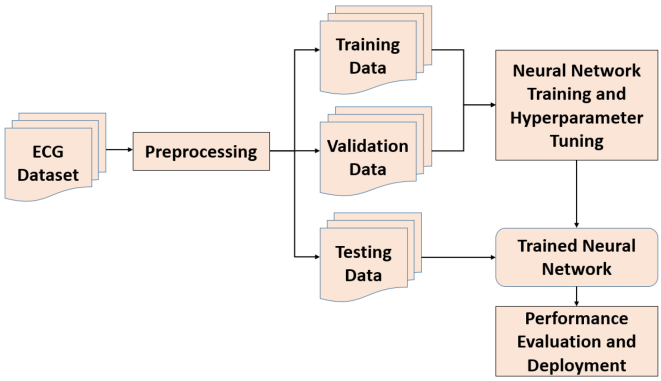



Fig. 2. Development flow for neural network-based ECG classification.

- *Manual:* In this case, medical professionals record the ECG signal at the hospital and identify the abnormal beats by visual inspection. This requires frequent visits to the hospital which are time-consuming and inconvenient for most people. Moreover, arrhythmia may get detected late as there is no monitoring of the heart activity in the time span between successive hospital visits.
- *Semi-automated:* Problems such as inconvenience in frequent hospital visits and late arrhythmia detection in the manual approach can be solved by using wearable healthcare devices. Such devices allow the monitoring of heart activity without hospital visits. These devices contain sensors that can directly record the ECG signal and are also equipped with hardware that can identify the types of abnormal beats. If the hardware in such devices uses traditional machine learning techniques (which do not involve neural networks) like support vector machine [18], the features have to be first manually extracted from the ECG recording and provided as inputs to the device. Hence, such an approach is known as semi-automated. It suffers from poor classification performance due to the imprecise nature of manual feature extraction.
- *Fully automated:* The need for manual feature extraction in the semi-automated approach can be eliminated by using neural networks. They are inherently capable of extracting the features from ECG recordings and then performing classification based on the extracted features. Hence, this approach is called fully automated. Moreover, automatic feature extraction results in superior classification performance compared to manual feature extraction. This can prove crucial for correct diagnosis and timely treatment.

Hence, neural network-based fully automated ECG classification is the most effective approach to building smart arrhythmia detection solutions. The generic flow for the development of neural network-based ECG classification solutions is shown in Fig. 2 . The recorded ECG data is pre-processed to remove the noise and enhance the regions of interest such as the 'QRS' complex in each ECG beat. It is then divided into a training set, a validation set, and a test set. The neural network training and hyperparameter tuning is performed using the training set and validation set, respectively. The classification performance of the trained network is then evaluated using the test set followed by the model deployment once the performance on the test set is deemed satisfactory.

### C. Related Work

There are many works that achieve high accuracy (around 98-99%) by leveraging various complex neural network topologies such as LSTMs [8], BLSTMs [7], [10], CNNs [4], [9], [12] and hybrid networks which combine LSTMs with CNNs [11], [19]. However, they just focus on software model development without any consideration of hardware resource requirements. Hence, they end up with networks that provide high accuracy at the expense of large energy consumption. This is not desirable for personalized healthcare devices which are battery-powered as frequent charging due to fast draining of the battery makes continuous health monitoring impractical and cumbersome. Hence, there is a need for energy-efficient ECG classification where such energy efficiency is achieved while still maintaining high accuracy. Works like [6], [13] leverage spiking neural networks to provide energy efficiency but suffer from low accuracy. Alternatively, RRAM-based hardware for ECG classification using non-spiking neural network in [14] can be leveraged for energy efficiency. However, it does not even consider the full AAMI classes (just considers a subset and ignores many types of possible input heartbeats) and still suffers from low accuracy. This is because it performs ECG classification in a naive manner without any innovation at the architecture or neural network level. The network architecture in [5] allows some parts of the network to be deactivated when not needed. However, as it uses AAMI classes which have intermixing of arrhythmia types with different severity impacts, it cannot take advantage of the fact that more severe arrhythmia types occur rarely and its network parts cannot be deactivated for a significant amount of time, which makes the energy savings very small. Moreover, as the architecture in [5] internally uses large neural network components, its energy efficiency is reduced. The most energy-efficient ECG classification among state-of-the-art works is provided by [3] which exploits the characteristics of ECG data to achieve a high degree of computation reuse. However, as it also uses AAMI classes having intermixing of arrhythmia types with different severity impacts, it misses out on a potentially huge improvement in energy efficiency that can be achieved via severity-based classification. In this paper, we develop a highly energy-efficient ECG classification architecture by leveraging the difference in severity impact of arrhythmia classes, while still maintaining high accuracy.

## III. Proposed Severity-Based Classification

### A. Concept

A medical disorder represents improper functioning of a certain organ in the human body and has various subtypes based on the manner in which the malfunctioning occurs in that organ. For instance, arrhythmia is a disorder that indicates improper functioning of the heart, and there exist different arrhythmia subtypes based on which part of the heart is affected as well

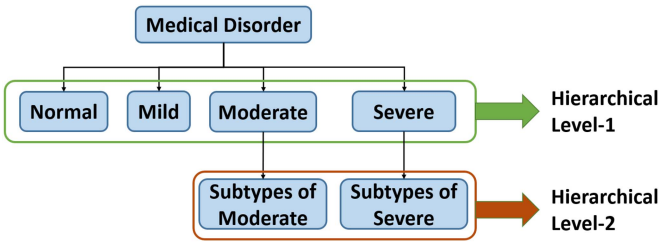



Fig. 3. Severity-based hierarchical classification for wearable healthcare devices intended to monitor any medical disorder.

as the way in which it is affected. The disorder subtypes differ in severity impact based on the extent to which they obstruct the organ's normal functioning. Severe subtypes may highly impact the organ leading to life-threatening situations, while others may have a minor impact leading to just a temporary inconvenience. The severity impact determines the urgency with which a person should seek medical help for certain disorder subtypes. For example, severe subtypes may need medical attention immediately while the non-severe ones may need it within a few days. Moreover, the desired speed of diagnosis and treatment is also governed by the severity impact. For instance, severe subtypes may need faster diagnosis and treatment to prevent further damage to the health over time, while such a speedup may not be necessary for the non-severe subtypes.

The influence of severity impact on the urgency in seeking medical attention as well as the speed of diagnosis and treatment can be leveraged to create severity-based classification with a two-level hierarchical structure as shown in Fig. 3. The first level of the hierarchy is intended for the user of the wearable device and indicates how urgently one needs to seek medical attention. Knowing the severity impact alone would suffice for this purpose and there is no need to know the exact disorder subtype. Hence, we can group the disorder subtypes into four broad classes based on their severity impact as follows:

- *Normal:* This class represents the normal working of the human body and the absence of the considered medical disorder. Hence, this category does not require any medical attention.
- *Mild:* This class includes disorder subtypes that have a very minor impact on normal organ functioning and do not lead to life-threatening scenarios over time. It is advisable to schedule a checkup in the upcoming few days if this class is detected.
- *Moderate:* This class includes disorder subtypes that have a minor impact on normal organ functioning at onset, but can potentially result in life-threatening scenarios over time. It requires medical attention much more quickly compared to the mild class, but not immediately.
- *Severe:* This class includes disorder subtypes that can significantly affect normal organ functioning at onset and are very likely to lead to life-threatening scenarios. It requires immediate medical attention upon detection.

Such a grouping can potentially improve the classification accuracy as the wearable device needs to detect only four broad classes instead of tens of subtypes of the considered medical disorder. Moreover, as the number of output classes is reduced, a smaller neural network can be used to reduce energy consumption while still maintaining high accuracy. The second level of this hierarchy is intended for speeding up the diagnosis and treatment. This can be achieved if the wearable device detects the exact disorder subtype and presents this information to the medical professional. Such speedup is only required for disorder subtypes that are either life-threatening from the onset or which become life-threatening over time. Hence, we need to only detect the disorder subtypes which are grouped together into moderate and severe classes at hierarchical level-1. As a result, hierarchical level-2 only consists of subtypes of moderate class and subtypes of severe class. We refer to the process of detecting the disorder types contained within a broad level-1 class as finer classification. As discussed earlier, it is clear that finer classification is only required for moderate and severe classes in level-1. Finer classification becomes redundant and is not required for mild class as everything will be thoroughly examined in a full checkup. Also, there is no need for finer classification of normal class as it needs no medical attention. As finer classification is not required for all disorder subtypes, this also simplifies the classification task as well as the hardware design providing further accuracy and energy efficiency benefits.

### B. ECG Classification

For this work, Table II shows the mapping of arrhythmia classes (subtypes) in the MIT-BIH arrhythmia dataset [15] to our severity-based classification structure. The rationale behind this mapping can be explained as follows [20], [21]:

- "N" beats belong to the normal class as they represent the normal working of the heart. Moreover, "paced" beats (/) also belong to the normal class as they indicate the normal working of the heart when aided by a pacemaker.
- "L," "R," "e," "j" beats belong to mild class because even though they deviate from perfectly normal beats ("N" and "paced"), they do not affect the functioning of the heart significantly and do not result in life-threatening scenarios over time.
- "A," "a" and "S" beats are related to improper functioning of atria which do not contribute significantly to the blood circulation process, while "J" and "E" beats indicate only a minor impact on ventricles which are vital for blood circulation. Hence, these beats have little impact on proper heart functioning at the onset. However, they can potentially lead to life-threatening scenarios over time and hence belong to the moderate class.
- "V" beat arises due to abnormal functioning of ventricles which are vital for blood circulation and thus indicates a danger to human life. "F" and "f" beats represent superimposition of cardiac cell potentials which can also lead to life-threatening scenarios. Hence, "V," "F" and "f" belong to the severe class. Moreover, we conservatively include the unclassifiable beat ("Q") in the severe class as its exact nature is not clear.

Human health falls into normal and mild classes much more often compared to moderate and severe classes. When the health

TABLE II
SEVERITY-BASED ECG CLASSIFICATION HIERARCHY FOR ARRHYTHMIA CLASSES IN MIT-BIH DATASET WITH DETAILS REGARDING MEDICAL ATTENTION AND FINER CLASSIFICATION

| Class in Hierarchical Level-1 | Arrhytmia Class in MIT-BIH Dataset | Advice for Medical Attention | Finer Classification |
|---|---|---|---|
| Normal | Normal Beat (N) | No medical attention required. | Not required. |
| | Paced Beat (/) | | |
| Mild | Left Bundle Branch Block Beat (L) | Schedule a heart checkup in the upcoming days. | Not required. |
| | Right Bundle Branch Block Beat (R) | | |
| | Atrial Escape Beat (e) | | |
| | Nodal (junctional) Escape Beat (j) | | |
| Moderate | Atrial Premature Beat (A) | Seek medical attention within a few hours. | Required. |
| | Aberrated Atrial Premature Beat (a) | | |
| | Nodal (junctional) Premature Beat (J) | | |
| | Supraventricular Premature Beat (S) | | |
| | Ventricular Escape Beat (E) | | |
| Severe | Fusion of Ventricular and Normal Beat (F) | Seek medical attention immediately. | Required. |
| | Premature Ventricular Contraction (V) | | |
| | Fusion of Paced and Normal Beat (f) | | |
| | Unclassifiable Beat (Q) | | |

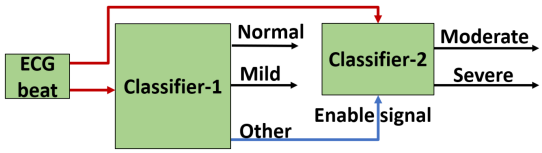



Fig. 4. An example of a hierarchical classification architecture.

is in the moderate or severe class, people seek medical attention, and the health returns back to normal or mild class. Thus, the ECG classifier deals with normal and mild classes much more frequently compared to moderate and severe classes. This fact can be leveraged for achieving energy efficiency by using a hierarchical connection of small classifiers which deal with only a subset of arrhythmia classes and are activated only when necessary. For example, consider the hierarchical architecture shown in Fig. 4. Both classifier-1 and classifier-2 are provided with the same ECG input. Initially, only classifier-1 is activated. If the input does not fall into the normal or mild class, classifier-1 activates classifier-2 which then classifies the input into the moderate or severe class. Classifier-2 remains off for most of the time as the inputs from moderate and severe classes occur less frequently, leading to energy savings. Moreover, as classifier-1 is always active, simplifying its design (e.g. using a small neural network) can also bring a reduction in energy consumption. This concept can be leveraged to design an energy-efficient hierarchical hardware architecture as described in the next section.

## IV. PROPOSED HIERARCHICAL HARDWARE ARCHITECTURES

The hierarchical hardware architecture can achieve energy efficiency by

- Activating the classifiers associated with infrequently occurring classes only when needed.
- Using simpler/smaller neural networks for classifiers dealing with frequently occurring classes.

This approach leads to four possible hardware architectures for the severity-based ECG classification which are shown in Table III , discussed next.

### A. Architecture-1

This is the simplest architecture for the severity-based ECG classification. Classifier-1 classifies the input into four classes: normal, mild, moderate and severe. Classifier-2 and classifier-3 classify moderate and severe classes further into their subtypes. Classifier-1 activates classifier-2 or classifier-3 when it detects moderate or severe class.

This architecture leads to energy savings as classifier-2 and classifier-3 remain inactive for most of the time. As classifier-1 is always on, it needs to use a smaller neural network to improve energy efficiency. Moreover, high accuracy for classifier-1 is important as its output advises the user about seeking medical help. However, a small neural network may not lead to high accuracy for classifier-1. Hence, there is a potential challenge of simultaneously achieving high accuracy and energy efficiency for classifier-1 in this architecture.

### B. Architecture-2

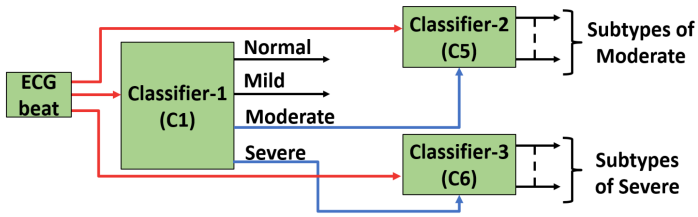
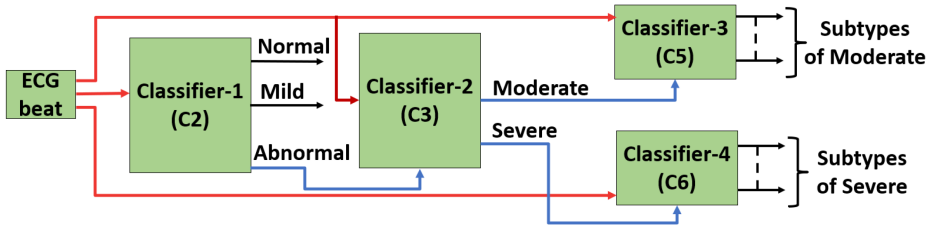
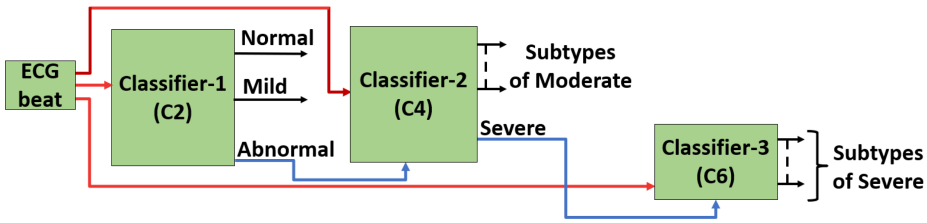
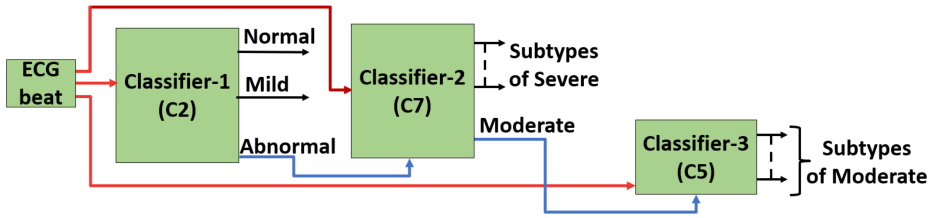
Architecture-2 can facilitate high accuracy with a small neural network for classifier-1 by using only three classes. This is achieved by grouping moderate and severe classes into a single abnormal class for classifier-1 and using an additional classifier-2 to split the abnormal class into moderate and severe classes. Classifier-2 activates classifier-3 and classifier-4 to classify moderate and severe classes further into their subtypes.

This architecture can potentially achieve high accuracy and low energy consumption for classifier-1 by using a smaller neural network, as classifier-1 now handles three classes unlike four classes in architecture-1. However, this architecture requires a total of four classifiers instead of three classifiers in architecture-1 which can increase overall energy consumption.

### C. Architecture-3

Architecture-3 can reduce the number of classifiers from four (in architecture-2) to three, while still maintaining only three classes in classifier-1. This is achieved by making classifier-2 handle six classes: five of them being subtypes of moderate (A, a, J, S, E) and sixth being the severe class. Thus, classifier-1 still handles three classes: normal, mild, and abnormal. It

TABLE III
POSSIBLE HARDWARE ARCHITECTURES FOR SEVERITY-BASED ECG CLASSIFICATION. ECG INPUT DATA IS INDICATED BY RED ARROWS, WHILE THE BLUE ARROWS REPRESENT CLASSIFICATION OUTPUTS WHICH ALSO ACT AS THE ENABLE SIGNALS

| Name | Architecture Structure |
|---|---|
| Architecture-1 |   |
| Architecture-2 |   |
| Architecture-3 |   |
| Architecture-4 |   |

Classifier names are shown in brackets, where the classifiers present in different architectures but having the same output classes are given the same name.

activates classifier-2 if it detects abnormal class. Classifier-2 then classifies abnormal class into six classes: A, a, J, S, E, and severe. If classifier-2 detects severe class then it activates classifier-3 which further classifies the severe class into its subtypes.

This architecture can retain all the benefits of classifier-1 in architecture-2 while reducing the overall energy consumption compared to architecture-2 as it needs only three total classifiers unlike four total classifiers in architecture-2. However, classifier-2 in this architecture has to deal with six classes unlike classifier-2 in architecture-2 which deals with only two classes. This can result in reduced accuracy.

### D. Architecture-4

Architecture-4 provides another way of reducing the total number of classifiers in architecture-2 by making classifier-2 handle five classes: four of those being subtypes of severe (F, V, f, Q) and the fifth one being the moderate class. Classifier-1 still has to deal with only three classes: normal, mild, abnormal and activates classifier-2 if it detects the abnormal class. Classifier-2 then classifies abnormal into five classes: F, V, f, Q, and moderate. If classifier-2 detects moderate class then it activates classifier-3 which further classifies moderate into its subtypes.

TABLE IV
LIST OF CLASSIFIER COMPONENTS REQUIRED FOR VARIOUS ARCHITECTURES

| Classifier | Output Classes | Used in |
|---|---|---|
| C1 | Normal, Mild, Moderate, Severe | Arch1 |
| C2 | Normal, Mild, Abnormal | Arch2, Arch3 |
| C3 | Moderate, Severe | Arch2 |
| C4 | Severe and Moderate Subtypes: A, a, J, S, E | Arch3 |
| C5 | Moderate Subtypes: A, a, J, S, E | Arch1, Arch2 |
| C6 | Severe Subtypes: F, V, f, Q | Arch1, Arch2, Arch3 |

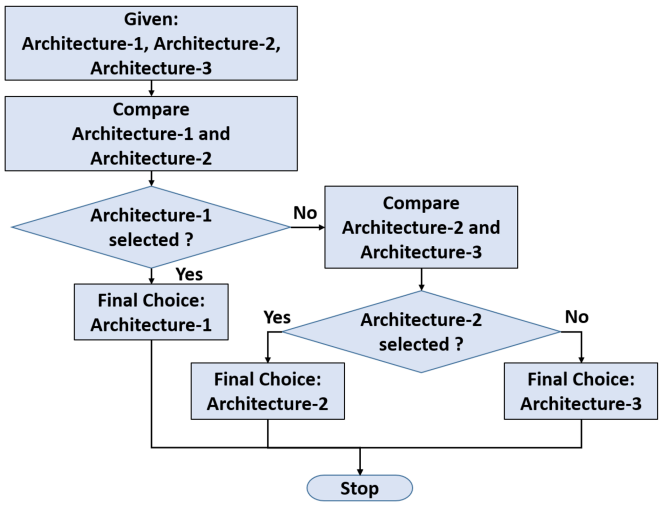



Fig. 5. Architecture selection process.

Classifier-1 in this architecture provides the same benefits as classifier-1 in architecture-3. However, classifier-2 and classifier-3 remain on for significantly more amount of time compared to those in architecture-3 as moderate classes occur more frequently than severe ones. This makes architecture-4 a worse version of architecture-3 in terms of energy efficiency. Hence, we do not select architecture-4, and it is just included here for completeness purpose.

### *E. Architecture Selection Criteria*

Different classification architectures lead to different classification accuracy and energy consumption. We can rule out architecture-4 as discussed in the previous subsection which leaves us with architecture-1, architecture-2, and architecture-3 as possible choices. Our goal now is to develop a methodology for selecting the appropriate architecture out of these three options. Table IV lists various classifier components needed for implementing architecture-1, architecture-2, and architecture-3. We assign names (C1, C2....C6) to the individual classifier components to make it easy to refer to a particular classifier. The architecture selection process consists of two phases as depicted in Fig. 5. In the first phase, our goal is to select an architecture that results in more number of timely hospital visits for the user of the wearable healthcare device. This means the selected architecture should have more accuracy on abnormal (moderate and severe) classes which is governed by classifier C1 for architecture-1 and classifier C2 for architecture-2 as well as architecture-3. Thus, if classifier C1 is better at detecting abnormal classes then we select architecture-1. Otherwise, we discard architecture-1 if classifier C2 performs better on abnormal classes and perform further exploration to select either architecture-2 or architecture-3 in the second phase.

The selection between architecture-1 and architecture-2 in the first phase depends on the choice between C1 and C2 as follows:

- If C1 has a significantly higher accuracy compared to C2, then architecture-1 should be selected.
- If C2 turns out to be significantly more accurate than C1, then architecture-2 should be chosen.
- If C1 and C2 have similar overall accuracy, then select architecture-1 if C1 has better accuracy on moderate and severe classes, otherwise select architecture-2 if C2 has better accuracy on moderate and severe classes. The selection is based on moderate and severe classes as they can lead to life-threatening scenarios.
- If C1 and C2 have similar overall accuracy as well as similar accuracy on moderate and severe classes, select the architecture containing the classifier which consumes less energy.

If we end up selecting architecture-2, we subsequently explore the choice between architecture-2 and architecture-3 in the second phase. This selection depends on the comparison between C3 and C4 as follows:

- If C3 turns out to be significantly more accurate than C4, we select architecture-2.
- If C4 has much higher accuracy than C3, we select architecture-3.
- If C3 and C4 have similar overall accuracy, then select architecture-2 if C3 has higher accuracy on severe class, otherwise select architecture-3 if C4 has higher accuracy on severe class. The decision is based on the severe class here as it can result in life-threatening situations.
- If C3 and C4 have similar overall accuracy and similar accuracy on severe class, select the one containing the classifier which consumes less energy.

Thus, we have defined clear selection criteria for the architectures based on the comparison of associated classifiers. In order to use these selection criteria, we have to first determine the topology and network configuration for each classifier which provides the best balance between accuracy and energy efficiency when implemented in hardware. The next section describes the details of the hardware implementation of these classifiers using RRAM-based computation-in-memory, followed by design space exploration to find out the network topology and configuration for the best balance between accuracy and energy efficiency.

## V. IMPLEMENTATION OF HIERARCHICAL HARDWARE ARCHITECTURES

Neural networks are conventionally implemented using hardware platforms like CPUs [22], GPUs [23], and AI-oriented ASICs like TPUs [24] which are based on the von-Neumann architecture and CMOS technology. The von-Neumann architecture entails the physical separation of memory and computation units which leads to a large number of data transfers to execute

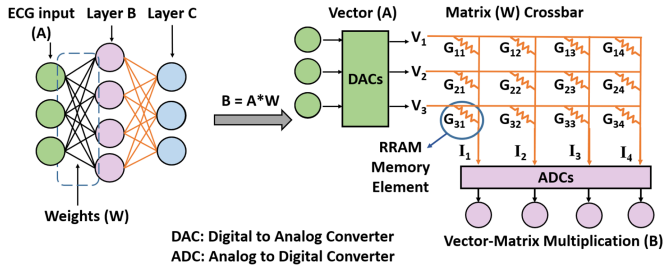
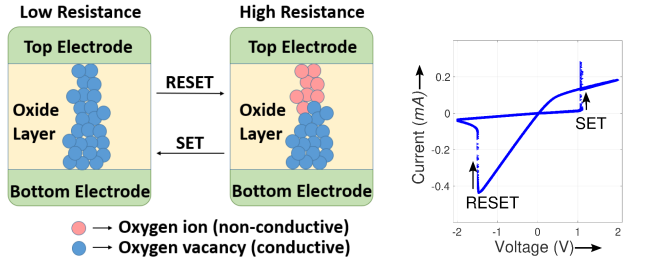



Fig. 6. Vector-matrix multiplication for ECG classification neural network using Computation-In-Memory (CIM).




(a) SET and RESET processes in RRAM. (b) RRAM I-V characteristics.

Fig. 7. RRAM device technology.

*vector-matrix multiplication* (VMM) operations for neural network applications. This results in high energy consumption and loss of performance as VMM operations account for more than 75% of the computations in neural networks [25], [26], [27], [28], [29], [30] Moreover, CMOS technology is facing issues like excessive sub-threshold leakage and scalability challenges [31], [32], [33]. *Computation-in-memory* (CIM) utilizes emerging memory technologies such as *resistive random access memories* (RRAMs) [34], [35], [36], [37], [38] to overcome the aforementioned limitations of the von-Neumann architecture and CMOS technology. Data storage in the form of RRAM conductance allows CIM to leverage circuit laws (Ohm's law and Kirchhoff's current law) to perform computing within the memory itself which eliminates the data transfer bottleneck. Moreover, RRAM devices overcome the technological challenges faced by CMOS as they are non-volatile (leakage-free), highly scalable, and small in size. Thus, CIM becomes a promising alternative to the conventional hardware for neural networks [39], [40], [41], [42]. Hence, we select the RRAM-based CIM paradigm to implement the neural networks required for the ECG subclassifiers.

### A. RRAM-Based Computation-in-Memory for ECG Classification

*1) Computation-in-Memory (CIM) Paradigm:* Mapping of VMM operation between two layers of an ECG classification neural network to CIM hardware is shown in Fig. 6. CIM uses memory elements that store the data in the form of conductance. A mesh-like structure built using such memory elements is called the crossbar. The crossbar performs computations in the analog domain and exchanges data with other digital system components using data converters like digital-to-analog converters (DACs) and analog-to-digital converters (ADCs). Weights are mapped to conductances ($G$'s) in the crossbar and ECG inputs are applied in the form of voltages ($V$'s) using DACs. The resulting current through all the $G$'s due to Ohm's law is equivalent to element-wise multiplication of $V$'s and $G$'s. The accumulation of currents from $G$'s in the same column due to Kirchhoff's law gives the accumulation of the element-wise products in the form of output currents ($I$'s). Thus, CIM performs a multiply-and-accumulate operation in the analog domain for each column. The multiply-and-accumulate operations across all the columns constitute a VMM operation. Thus, VMM is performed with O(1) time complexity as all the columns produce the outputs at the same time. ADCs then convert the column currents to digital outputs. The digitized VMM output is sent to other parts of the system for further operations.

*2) RRAM Device Technology:* The RRAM device (also known as memristor) is made up of an oxide material sandwiched between two metal electrodes. It has a *high-resistance state* (HRS) and a *low-resistance state* (LRS). These states can be used to store data as 0 or 1. The transition from HRS to LRS is called "SET", whereas that from LRS to HRS is called "RESET". When a set voltage ($V_{SET}$) is applied to an RRAM device in HRS, it creates a conductive path called filament. This increases the conductivity of the oxide layer leading to a change of state from HRS to LRS. When reset voltage ($V_{RESET}$) is applied to an RRAM device in LRS, it causes rupture of the conductive filament. This reduces the oxide layer conductivity resulting in a change of state from LRS to HRS. Both SET and RESET processes for an RRAM device are depicted in Fig. 7. Reading the data from an RRAM device refers to detecting its resistance state. This is achieved by applying a very small voltage $V_{READ}$ ($V_{READ} << |V_{SET}|$ and $V_{READ} << |V_{RESET}|$) across it and measuring the resulting output current. A small output current means the device is in HRS and high output current means the device is in LRS. Moreover, a single RRAM device can exhibit more than two conductance states by controlling the extent of filament creation or rupture. This is known as a multi-level cell (MLC) operation which allows a single RRAM device to store multiple bits of information [43].

### B. Selection of Network Topology and Layer Configuration

After selecting the CIM paradigm for hardware implementation, the next task is to find the neural network (topology and layer configuration) for each classifier that provides the best balance between accuracy and energy consumption considering CIM-based hardware. This is achieved by design space exploration across four types of neural networks: fully-connected network (FC) [44], [45], long short-term memory network (LSTM) [46], [47], bidirectional long short-term memory network (BLSTM) [48], [49] and temporal convolutional network (TCN) [50], [51], as shown in Fig. 8. We first list the various classifiers needed for a given hierarchical hardware architecture. We then choose a classifier from this list and implement it using all four aforementioned network types (FC, LSTM, BLSTM and TCN) so that each network achieves its maximum possible accuracy. Energy consumption for all four resulting networks is then estimated by considering a CIM-based

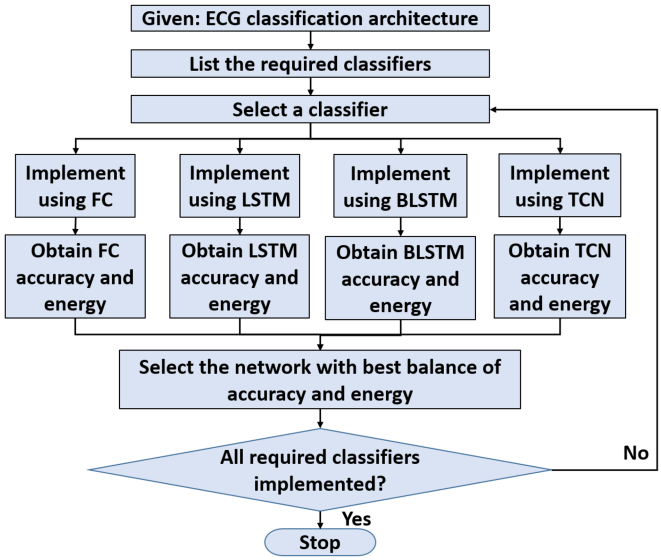



Fig. 8. Design space exploration flow for implementing a given hierarchical hardware architecture.

hardware implementation. Finally, we select the network (FC or LSTM or BLSTM or TCN) which provides the best balance between accuracy and energy consumption for the classifier. This process is repeated for all the classifiers required in the given architecture. Standard convolutional neural network (CNN) [52] is not included for this design space exploration as we already consider TCN, which is an advanced form of CNN that deals more effectively with time series data like ECG. For completeness, we will compare our hierarchical ECG classification with state-of-the-art CNN-based ECG classification in Section VII.

## VI. Simulation Setup

### A. ECG Dataset

We use the MIT-BIH arrhythmia dataset [15] available in Physiobank [16] for our simulation experiments. It consists of 30 minutes of ECG recording from 48 patients across 15 arrhythmia types. The distribution of ECG beats for each of the arrhythmia types is given in Table V. As this work focuses on ECG classification and not on QRS peak detection, we directly use the QRS peak annotations available in the MIT-BIH dataset. QRS peak detection at runtime can be achieved by algorithms like Pan-Tompkins algorithm [53] which can also be implemented in hardware [54].

### B. Performance Metrics

Performance metrics for the evaluation of the proposed hierarchical ECG classification are considered at two levels: algorithmic and hardware. Algorithmic performance metrics are accuracy and critical accuracy while the hardware performance metrics are energy and area. These are described in detail as follows:

*1) Algorithmic Metrics:*

- *Accuracy:* It is the ratio of the total number of correctly classified beats to the total number of input beats.

TABLE V
Distribution of ECG Beats in MIT-BIH Arrhythmia Dataset

| MIT-BIH Class | No. of Beats |
|---|---|
| Normal Beat (N) | 75022 |
| Paced Beat (/) | 7025 |
| Left Bundle Branch Block Beat (L) | 8072 |
| Right Bundle Branch Block Beat (R) | 7255 |
| Atrial Escape Beat (e) | 16 |
| Nodal (junctional) Escape Beat (j) | 229 |
| Atrial Premature Beat (A) | 2546 |
| Aberrated Atrial Premature Beat (a) | 150 |
| Nodal (junctional) Premature Beat (J) | 83 |
| Supraventricular Premature Beat (S) | 2 |
| Ventricular Escape Beat (E) | 106 |
| Fusion of Ventricular and Normal Beat (F) | 802 |
| Premature Ventricular Contraction (V) | 7129 |
| Fusion of Paced and Normal Beat (f) | 982 |
| Unclassifiable Beat (Q) | 33 |
| **Total** | 109452 |

TABLE VI
Critical Class Definitions for Classifiers

| Classifier | Output Classes | Critical Classes |
|---|---|---|
| C1 | Normal, Mild, Moderate, Severe | Moderate, Severe |
| C2 | Normal, Mild, Abnormal | Abnormal |
| C3 | Moderate, Severe | Severe |
| C4 | Severe and Moderate Subtypes: A, a , J, S, E | Severe |
| C5 | Moderate subtypes: A, a, J, S, E | - |
| C6 | Severe subtypes: F, V, f, Q | - |

- *Critical Accuracy:* We define critical classes as a subset of the total output classes that can be more life-threatening and hence considered more important. Table VI defines critical classes for various classifiers required in severity-based classification architectures presented in Section IV. Please note that the concept of critical classes is only applicable to classifiers that handle at least one of the broad classes (normal, abnormal, mild, moderate, and severe) which are fundamentally based on severity differences. It is not applicable to classifiers that only handle subtypes of moderate or subtypes of severe as the subtypes indicate similar severity levels. *We now define critical accuracy as the ratio of the total number of correctly classified beats belonging to the critical classes to the total number of input beats belonging to the critical classes.* For instance, consider classifier C1 in Table VI with four output classes: normal, mild, moderate, and severe. As moderate and severe can lead to life-threatening scenarios, they are considered critical classes. Critical accuracy for classifier C1 can then be obtained as follows:

$\text{Correct}_{\text{crit}}$ : Correct classified moderate and severe beats

$\text{Total}_{\text{crit}}$ : Total input moderate and severe beats

$$\text{Critical accuracy for C1 (in \%)} = 100 \times \frac{\text{Correct}_{\text{crit}}}{\text{Total}_{\text{crit}}}$$

*2) Hardware Metrics:*

- *Energy:* As wearable healthcare devices are battery-powered, the energy consumed by the CIM-based ECG classifier is an important hardware performance metric.
- *Area:* Apart from energy efficiency, wearable healthcare devices should be compact in size. Hence, the area occupied by the CIM-based ECG classifier is considered another hardware performance metric.

### C. Simulation Platform

Accuracy and critical accuracy are evaluated by implementing the neural networks using PyTorch [55] with RMSProp [56] optimizer. The details of the used neural networks are described below. Please note that only the number of output neurons ($n_{out}$) varies from two to six based on which classifier is being implemented, the rest stays the same.

- *Fully-connected network (FC) [44], [45]:* It has an input layer of 250 neurons, a hidden layer of 100 neurons, and $n_{out}$ output neurons. FC network can thus be expressed as 250-100-$n_{out}$. The activation function used is ReLU.
- *Long short-term memory network (LSTM) [46], [47]:* An input sequence of 250 samples is fed to two cascaded standard LSTM units, each having a hidden state size of 30. The output from the last LSTM unit corresponding to the final timestep is flattened and connected to an output layer consisting of $n_{out}$ neurons. LSTM structure can be expressed as 250-LSTM(30)-LSTM(30)-Flatten-$n_{out}$.
- *Bidirectional long short-term memory network (BLSTM) [48], [49]:* Its structure is exactly the same as the LSTM described earlier, with standard LSTM units being replaced by their bidirectional version. BLSTM structure can be expressed as 250-BLSTM(30)-BLSTM(30)-Flatten-$n_{out}$.
- *Temporal convolutional network (TCN) [50], [51]:* It is provided with a 250 sample long single channel input sequence. This sequence is fed into a cascade of six temporal blocks. The convolutions within each temporal block have a kernel size of four and 20 output channels. Output from the last temporal block corresponding to the final timestep is flattened and connected $n_{out}$ neurons in the output layer. TCN structure can be expressed as 250-TB1-TB2-TB3-TB4-TB5-TB6-Flatten-$n_{out}$, where TBn represents $n^{th}$ temporal block.

We split the ECG data as 60% for the training set, 20% for the validation set, and 20% for the test set. The networks are trained using the training set and the validation set is used for hyperparameter tuning. The test set is not exposed to the network during training or the hyperparameter tuning process. It is used only after the network is fully trained and tuned. All the accuracy and critical accuracy results are presented for the test set so that they correctly reflect the generalization performance on unseen test data.

We have developed a Python-based framework to estimate energy and area for neural networks using computation-in-memory hardware known as ISAAC presented in [57]. Its main building block is shown in Fig. 9. The full-precision neural network weights and inputs are split into smaller bit-size chunks called slices. This is because i) the bit-capacity of RRAM devices is typically less than bit-size needed for neural network weights and ii) digital-to-analog converters (DACs) and analog-to-digital converters (ADCs) with high bit-resolutions consume high energy and area. For example, as shown in Fig. 9, 2-bit slices of the weights are converted to conductances and mapped to RRAMs in different crossbar columns, while 1-bit slices of the inputs are converted to voltages and mapped to different time-steps in which they are applied to the crossbar. With 1-bit DACs and 16-bit digital inputs as an example, 1-bit is fed at a time to all the DACs and this process is repeated 16 times (called 16 timesteps). The DACs convert the bits into equivalent voltage which produces a current at the output of every column in the crossbar. These currents are latched into sample and hold circuits (S&H) and then converted to digital outputs by ADCs. The outputs of ADCs belong to various weight slices based on which column they come from, and to different input data slices based on which timestep they belong to. To account for the slicing of weights across different crossbar columns, ADC outputs undergo shift and add operations across columns. Moreover, to account for time-multiplexed inputs (1-bit at a time), the shifted and added ADC outputs undergo another round of shift and add operations for merging with the outputs from previous timesteps to produce the full-precision digital output. We estimate the energy and area for neural networks using the design in [57] which utilizes this functionality.

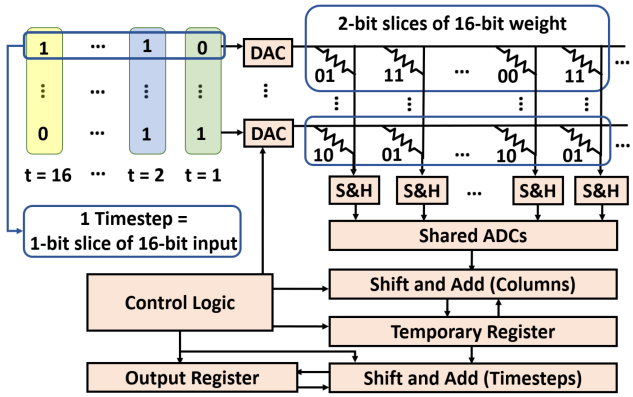
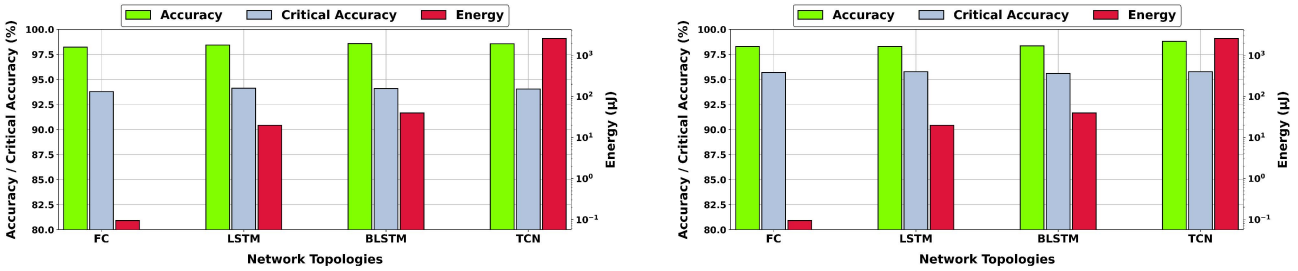



Fig. 9. Computation-in-memory design for vector-matrix multiplication.

### D. Experiments Performed

*1) Architecture Selection:* The goal of this experiment is to select the hierarchical classification architecture which provides the best balance between accuracy and energy efficiency, where architecture-1, architecture-2, and architecture-3 are the possible choices. We leverage the criteria described in Section IV-E for this selection and use design space exploration described in Section V-B to obtain the neural network types and configurations for classifiers in the selected architecture.

*2) Comparison With State-of-The-Art:* In this experiment, we demonstrate the effectiveness of our hierarchical ECG classification by comparing its performance with state-of-the-art ECG classifiers based on metrics described in Section VI-B.




(a) Design space exploration for classifier C1

(b) Design space exploration for classifier C2

Fig. 10. Comparison of classifiers C1 and C2 for selection between architecture-1 and architecture-2.

TABLE VII
COMPARISON OF THE PROPOSED HIERARCHICAL CLASSIFICATION WITH STATE-OF-THE-ART ECG CLASSIFIERS. VALUES MARKED WITH * ARE ESTIMATED BY OUR FRAMEWORK ASSUMING RRAM-BASED CIM IMPLEMENTATION

| Performance Metric | Wu-IEEE Access'2019 [3] | Xiao-JBHI'2022 [4] | Wang-TBCAS'2019 [5] | This Work |
|---|---|---|---|---|
| **Output Classes** | AAMI: N, S, V, F, Q | AAMI: N, S, V, F, Q | AAMI: N, S, V, F, Q | Severity-based: Normal, Mild, Moderate, Severe |
| **Finer Classification** | No | No | No | Yes |
| **Accuracy (%)** | 96.06 | 99.10 | 98.40 | 98.29 |
| **Energy per heartbeat classification ($\mu$J)** | 2.78 | 710.00 | 488.81* | 0.11 |
| **Area (mm$^2$)** | 1.40 | - | 1.11* | 0.11 |
| **Hardware Design Complexity** | High | High | High | Low |

Unavailable or not applicable values are indicated by "-".

## VII. SIMULATION RESULTS

### A. Architecture Selection

As discussed in Section IV-E, we break the task of selecting the appropriate architecture into two phases. In the first phase, we make a choice between architecture-1 and architecture-2. The selection process stops if architecture-1 is selected. Otherwise, we proceed to the second phase to make a selection between architecture-2 and architecture-3, as our final choice.

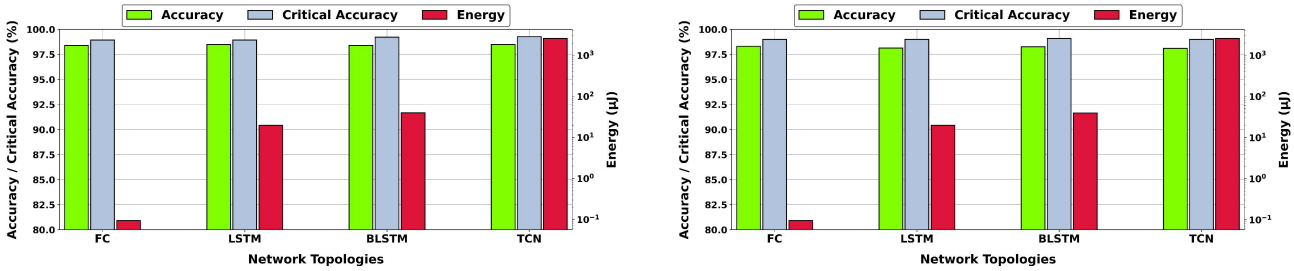
For the first phase, the choice between architecture-1 and architecture-2 is governed by the comparison of classifiers C1 (Normal vs Mild vs Moderate vs Severe) and C2 (Normal vs Mild vs Abnormal) described in Table IV. We implement both C1 and C2 using all four types of neural networks (FC, LSTM, BLSTM and TCN) as described in Section V-B. Fig. 10 shows the performance metrics across various network topologies for C1 and C2. It is clear that FC provides the best balance between accuracy and energy efficiency for both C1 and C2. FC achieves accuracy comparable to other network topologies (LSTM, BLSTM, TCN) as the classification boundaries for ECG data seem to be simple and do not benefit much from the extra computational powers in other topologies. The low energy consumption of FC can be attributed to two factors:

- It needs fewer hardware resources compared to LSTM, BLTSM, and TCN as it doesn't involve complex computations like hidden state updates in LSTM/BLSTM or convolution operation in TCN.
- LSTM, BLSTM, and TCN involve iterative computations such as updating the hidden state after each input sample (LSTM and BLSTM) or sliding convolution windows across input feature maps (TCN). Thus, they use the same hardware multiple times and total energy is the sum of energies required for each iteration. The energy consumption increases further as such iterative computation is needed for each layer in the network. FC just requires a single non-iterative matrix-matrix multiplication per layer, saving a lot of energy.

CNN [52] is not included in the above comparison as clarified in Section V-B. Nevertheless, CNN will also suffer from high energy consumption problem like TCN because both of them involve iterative sliding window convolution as the basic computation. We quantitatively demonstrate this later in Table VII by comparing our FC-based ECG classification with CNN-based state-of-the-art ECG classification [3], [4], [5].

Mixing network topologies into a hybrid structure can potentially yield better results when the topologies that are being mixed have a significant difference in accuracy but not a large difference in energy consumption. However, Fig. 10 shows that all of these networks deliver similar accuracy while LSTM, BLSTM, and TCN consume much more energy than that of FC. Hence, topology mixing is not useful as it would result in adding a large energy-consuming component to the FC network for almost no change in accuracy.

As shown in Fig. 10, FC version of C2 achieves 2% higher critical accuracy (please see Table VI for critical classes) than the FC version of C1. This is because combining the moderate and severe classes in C1 into a single abnormal class in C2 simplifies the classification task, as C2 has to learn only three classification boundaries (normal vs mild vs abnormal) unlike four classification boundaries in C1 (normal vs mild vs moderate vs severe). Higher critical accuracy also indicates that C2 correctly detects more scenarios where the user needs to seek medical help. Hence, we select the FC version of C2 and thereby architecture-2 in the first phase.




(a) Design space exploration for classifier C3

(b) Design space exploration for classifier C4

Fig. 11. Comparison of classifiers C3 and C4 for selection between architecture-2 and architecture-3.

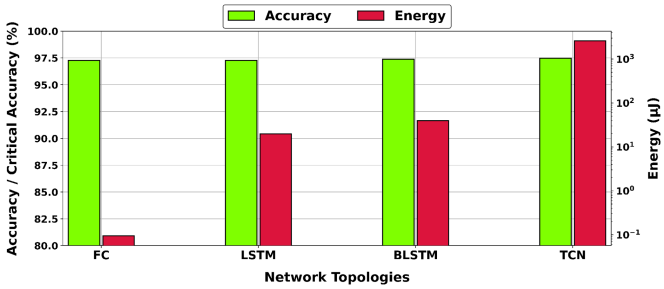



Fig. 12. Design space exploration for classifier C6.

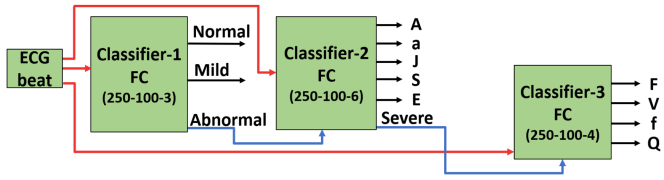



Fig. 13. Final classifier architecture with network type and configuration annotated.

Having selected architecture-2 in the first phase, we then begin the second phase to make a selection between architecture-2 and architecture-3. This depends on the comparison between classifiers C3 (Moderate vs Severe) and C4 (A vs a vs J vs S vs E vs Severe) described in Table IV. Both C3 and C4 are implemented using all four types of neural networks (FC, LSTM, BLSTM, and TCN) and their performance comparison across various network topologies is shown in Fig. 11. FC network ends up delivering the best balance between accuracy and energy efficiency for both C3 and C4, for the same reasons as discussed while comparing C1 and C2. It can also be seen that FC versions of C3 and C4 have almost identical performance across all the metrics. However, C4 leads to architecture-3 with a total of three classifiers while C3 leads to architecture-2 which needs a total of four classifiers. Thus, we select C4 (its FC version) and thereby architecture-3 as it needs fewer hardware resources and less energy, without any impact on accuracy and critical accuracy.

After finalizing architecture-3, the only remaining thing is to figure out the neural network type and configuration to use for its remaining classifier (classifier C6 in Table IV) which deals with the classification of subtypes of severe class (F, V, f, and Q). We implement it using all four possible types of neural networks (FC, LSTM, BLSTM, and TCN) and show their performance comparison in Fig. 12. The FC version is selected as it provides the best balance between accuracy and energy efficiency. Thus, our final selection is architecture-3 with all of its classifiers being FC networks with configurations as shown in Fig. 13. Classifier-1, classifier-2 and classifier-3 achieve accuracy of 98.29%, 98.31%, 97.26% and energy consumption of 0.094 $\mu$J, 0.095 $\mu$J, 0.094 $\mu$J, respectively. As the selection between architecture-1 and architecture-2 depends on the choice between classifiers C1 and C2 only, while that between architecture-2 and architecture-3 depends on the comparison between classifiers C3 and C4 only, we have not evaluated the accuracy of classifier C5 in Table VI. Moreover, as we end up selecting architecture-3 which does not include C5, there is no further need to evaluate its accuracy.

### B. Comparison With State-of-The-Art and Discussion

Performance comparison of the proposed hierarchical ECG classification with state-of-the-art is presented in Table VII. It includes [4] which represents the most accurate ECG classification and [3] which represents the most energy-efficient ECG classification, for AAMI classes among state-of-the-art works as discussed in Section II-C. We also include [5] in Table VII because its architectural approach (selectively turning off some classification components) is close to our paper.

The reported accuracy of 98.29% for the proposed ECG classification architecture in Table VII is the accuracy of classifier-1 in Fig. 13. This is because classifier-1 classifies the ECG beats only into the broad severity classes similar to the state-of-the-art works which classify the ECG beats into broad AAMI classes only and do not detect the actual arrhythmia classes. The accuracy comparison in Table VII shows that we achieve classification accuracy on par with state-of-the-art ECG classification solutions. Even though the accuracies obtained by our work and state-of-the-art works are very similar, the classification boundaries addressed by our work are different than the state-of-the-art. For instance, the normal beat in MIT-BIH dataset belongs to AAMI class "N," while paced beat belongs to AAMI class "Q". However, both normal beat and paced beat in MIT-BIH dataset belong to the same "Normal" class in our severity-based ECG hierarchy. Thus, the accuracy results do not reflect a fair comparison. Hence, the emphasis should be on the

TABLE VIII
ENERGY CONSUMPTION AND TEST SET FRACTION FOR SEVERITY CLASSES

| Severity Class | Energy per heartbeat classification | Heartbeats in test set (N) | Fraction of test set (N ÷ 21,891) |
|---|---|---|---|
| Normal | 0.094 $\mu$J | 16410 | 0.75 |
| Mild | 0.094 $\mu$J | 3115 | 0.14 |
| Moderate | 0.19 $\mu$J | 577 | 0.03 |
| Severe | 0.28 $\mu$J | 1789 | 0.08 |

fact that our work achieves good accuracy on severity-based classes, rather than comparing the absolute accuracy values.

For a heartbeat that belongs to a broad severity class or a broad AAMI class, finer classification refers to detecting its actual arrhythmia class. For instance, once we classify a beat into the broad severity class "Moderate", then finer classification determines the arrhythmia class of that beat out of the five arrhythmia classes ("A", "a", "J", "S", and "E") contained within the broad "Moderate" class. More details about finer classification can be found in Section III-A. As shown in Table VII, only the proposed severity-based architecture provides such finer classification which can help doctors with faster diagnosis and treatment.

The energy consumption for various severity-based classes in our proposed hierarchical ECG classification architecture (Fig. 13) is shown in Table VIII. If the input beat falls into the "Normal" or "Mild" class, only classifier-1 is active which consumes 0.094 $\mu$J. If the input beat gets classified as the "Moderate" class, both classifier-1 and classifier-2 get utilized consuming a total of 0.094 $\mu$J + 0.095 $\mu$J = 0.19 $\mu$J. If the input beat is identified as the "Severe" class, all classifiers get utilized consuming a total of 0.094 $\mu$J + 0.095 $\mu$J + 0.094 $\mu$J = 0.28 $\mu$J which is the worst-case energy consumption for any single heartbeat in our architecture. For a fair comparison with state-of-the-art works in Table VII which report average energy consumption, we derive the average energy consumption for our architecture as the weighted average of the energy consumption across the severity classes. Here, the weight coefficients used for averaging indicate what fraction of total heartbeats in the test set (21,891) belongs to a specific severity class as shown in Table VIII. This results in an average energy consumption of 0.11 $\mu$J per heartbeat classification with a standard deviation of 0.052 $\mu$J.

Table VII shows that the proposed hierarchical ECG classifier consumes 25× less energy and 12× less area compared to the state-of-the-art while keeping the accuracy benefits intact. The energy savings can be attributed to the fact that hierarchical architecture simplifies the design, activates the hardware components only when necessary, and uses RRAM-based computation-in-memory which further improves energy efficiency. Area savings arise from the design simplification due to hierarchical architecture as well as the high scalability of RRAM devices.

We have presented an architecture for ECG classification which can be transformed into a hardware chip, where existing neural network algorithms are implemented as hardware components. Hence, the complexity comparison with state-of-the-art refers to the complexity of designing such a chip. The use of simple fully-connected (FC) neural network topology in our proposed architecture greatly simplifies the dataflow, storage of intermittent calculations, and control logic compared to complex network topologies in [3], [4], [5]. This results in the simplification of various chip design processes like placement, routing, and timing analysis resulting in faster chip development. Hence, our architecture greatly reduces the hardware design complexity compared to the state-of-the-art as shown in Table VII.

## VIII. CONCLUSION

We propose severity-inclusive, accurate, and energy-efficient ECG classification using hierarchical hardware architecture and RRAM-based computation-in-memory (CIM) paradigm. The hierarchical structure achieves high accuracy by breaking down the complete classification task into smaller subtasks and saves energy by activating internal components only when they are needed. The hierarchical structure also accounts for severity differences between various arrhythmia classes to help the users of the wearable healthcare device in seeking timely medical attention as well as assist medical professionals with faster diagnosis and treatment. We further perform design space exploration to implement the internal components for the classification subtasks using the neural networks (topology and configuration) which provide high energy efficiency while still maintaining good accuracy, considering RRAM-based CIM hardware. The proposed ECG classification architecture achieves 25× improvement in terms of average energy consumption and 12× improvement in terms of area compared to the state-of-the-art. This work has shown that by smartly designing the computation architecture based on the characteristics of the application as well as the underlying hardware technology, one can achieve significant improvements in terms of energy efficiency and area footprint.

## ACKNOWLEDGMENT

The authors would like to thank Vasanti Thote (M.D.) for the in-depth and insightful discussions on cardiac arrhythmia.

## REFERENCES

[1] M. Naghavi et al., "Global, regional, and national age-sex specific mortality for 264 causes of death, 1980–2016: A systematic analysis for the global burden of disease study 2016," *Lancet*, vol. 390, no. 10100, pp. 1151–1210, 2017.
[2] "Heart disease and stroke statistics At-a-Glance," 2022. [Online]. Available: https://www.heart.org/idc/groups/ahamah-public/,wcm/,sop/,smd/documents/downloadable/ucm_470704.pdf
[3] J. Wu, F. Li, Z. Chen, Y. Pu, and M. Zhan, "A neural network-based ECG classification processor with exploitation of heartbeat similarity," *IEEE Access*, vol. 7, pp. 172774–172782, 2019.
[4] J. Xiao et al., "ULECGNet: An ultra-lightweight end-to-end ECG classification neural network," *IEEE J. Biomed. Health Inform.*, vol. 26, no. 1, pp. 206–217, Jan. 2022.
[5] N. Wang, J. Zhou, G. Dai, J. Huang, and Y. Xie, "Energy-efficient intelligent ECG monitoring for wearable devices," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 5, pp. 1112–1121, Oct. 2019.
[6] F. C. Bauer, D. R. Muir, and G. Indiveri, "Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor," *IEEE Trans. Biomed. Circuits Syst.*, vol. 13, no. 6, pp. 1575–1582, Dec. 2019.
[7] 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*, vol. 8, pp. 71189–71198, 2020.

[8] O. Yildirim et al., "A new approach for arrhythmia classification using deep coded features and LSTM networks," *Comput. Methods Programs Biomed.*, vol. 176, pp. 121–133, 2019.
[9] O. Yildirim et al., "Arrhythmia detection using deep convolutional neural network with long duration ECG signals," *Comput. Biol. Med.*, vol. 102, pp. 411–420, 2018.
[10] O. Yildirim, "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification," *Comput. Biol. Med.*, vol. 96, pp. 189–202, 2018.
[11] J. H. Tan et al., "Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals," *Comput. Biol. Med.*, vol. 94, pp. 19–26, 2018.
[12] Y.-J. Lin et al., "Artificial intelligence of things wearable system for cardiac disease detection," in *Proc. IEEE Int. Conf. Artif. Intell. Circuits Syst.*, 2019, pp. 67–70.
[13] F. Corradi et al., "ECG-based heartbeat classification in neuromorphic hardware," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.
[14] A. M. Hassan, A. F. Khalaf, K. S. Sayed, H. H. Li, and Y. Chen, "Real-time cardiac arrhythmia classification using memristor neuromorphic computing system," in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2018, pp. 2567–2570.
[15] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Eng. Med. Biol. Mag.*, vol. 20, no. 3, pp. 45–50, May/Jun. 2001.
[16] A. L. Goldberger et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
[17] "Recommended practice for testing and reporting performance results of ventricular arrhythmia detection algorithms," Association for the Advancement of Medical Instrumentation, Arlington, VA, USA, 1987.
[18] R. Thilagavathy et al., "Real-time ECG signal feature extraction and classification using support vector machine," in *Proc. IEEE Int. Conf. Contemporary Comput. Appl.*, 2020, pp. 44–48.
[19] P. Zhang et al., "Clsassification of ECG signals based on LSTM and CNN," in *Proc. Int. Conf. Artif. Intell. Secur.*, 2020, pp. 278–289.
[20] V. Fuster et al., *Hurst's The Heart*, 14th ed. New York, NY, USA: McGraw Hill, 2017.
[21] J. Jameson et al., *Harrison's Manual of Medicine*, 20th ed. New York, NY, USA: McGraw Hill, 2020.
[22] "Intel processor family," 2022. [Online]. Available: https://www.intel.in/content/www/in/en/products/processors/core.html
[23] "NVIDIA turing architecture GPUs," 2022. [Online]. Available: https://www.nvidia.com/en-in/geforce/turing/
[24] N. P. Jouppi et al., "In-datacenter performance analysis of a tensor processing unit," in *Proc. ACM/IEEE 44th Annu. Int. Symp. Comput. Architecture*, 2017, pp. 1–12.
[25] A. Sebastian et al., "Memory devices and applications for in-memory computing," *Nature Nanotechnol.*, vol. 15, pp. 529–544, 2020.
[26] A. Singh et al., "Referencing-in-array scheme for RRAM-based CIM architecture," in *Proc. Des., Automat. Test Europe Conf. Exhib.*, 2022, pp. 1413–1418.
[27] F. Cai et al., "A fully integrated reprogrammable memristor–CMOS system for efficient multiply-accumulate operations," *Nature Electron.*, vol. 2, no. 7, pp. 290–299, 2019.
[28] L. Ni et al., "Distributed in-memory computing on binary RRAM crossbar," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 13, no. 3, pp. 1–18, 2017.
[29] S. Shukla et al., "A scalable multi-TeraOPS core for AI training and inference," *IEEE Solid-State Circuits Lett,*, vol. 1, no. 12, pp. 217–220, Dec. 2018.
[30] S. Hamdioui et al., "Memristor for computing: Myth or reality?," in *Proc. Des., Automat. Test Europe Conf. Exhib.*, 2017, pp. 722–731.
[31] S. Borkar et al., "Design and reliability challenges in nanometer technologies," in *Proc. 41st Annu. Des. Automat. Conf.*, 2004, pp. 75–75, doi: 10.1145/996566.996588.
[32] N. Z. Haron and S. Hamdioui, "Why is CMOS scaling coming to an END?," in *Proc. 3rd Int. Des. Test Workshop*, 2008, pp. 98–103.
[33] "International Roadmap for Devices and Systems," 2022. [Online]. Available: standards.ieee.org/develop/indconn/irds/index.html
[34] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature Mater.*, vol. 18, pp. 309–323, 2019.
[35] L. Ni, Z. Liu, H. Yu, and R. V. Joshi, "An energy-efficient digital ReRAM-crossbar-based CNN with bitwise parallelism," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 3, pp. 37–46, Dec. 2017.
[36] S. Diware, A. Singh, A. Gebregiorgis, R. V. Joshi, S. Hamdioui, and R. Bishnoi, "Accurate and energy-efficient bit-slicing for RRAM-based neural networks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 1, pp. 164–177, Feb. 2023.
[37] A. P. James, "A hybrid memristor–CMOS chip for AI," *Nature Electron.*, vol. 2, no. 7, pp. 268–269, 2019.
[38] S. Hamdioui et al., "Memristor based computation-in-memory architecture for data-intensive applications," in *Proc. Des., Automat. Test Europe Conf. Exhib.*, 2015, pp. 1718–1725.
[39] P.-Y. Chen and S. Yu, "Technological benchmark of analog synaptic devices for neuroinspired architectures," *IEEE Des. Test*, vol. 36, no. 3, pp. 31–38, Jun. 2019.
[40] P.-Y. Chen, X. Peng, and S. Yu, "NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3067–3080, Dec. 2018.
[41] A. Siemon, D. Wouters, S. Hamdioui, and S. Menzel, "Memristive device modeling and circuit design exploration for computation-in-memory," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2019, pp. 1–5.
[42] S. Diware et al., "Unbalanced bit-slicing scheme for accurate memristor-based neural network architecture," in *Proc. IEEE 3rd Int. Conf. Artif. Intell. Circuits Syst.*, 2021, pp. 1–4.
[43] W. Kim et al., "Multistate memristive tantalum oxide devices for ternary arithmetic," *Sci. Rep.*, 2016, vol. 6, no. 1, pp. 1–9, 2016.
[44] Grosse, "CSC321 lecture 5: Multilayer perceptrons," 2018. [Online]. Available: https://www.cs.toronto.edu/rgrosse/courses/csc321_2018/slides/lec05.pdf
[45] "Multilayer perceptrons & neural networks: Basics, Kalev Kask," 2018. [Online]. Available: https://www.ics.uci.edu/kkask/Spring-2018%20CS273P/slides/08-mlpercept.pdf
[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[47] "LSTM understanding networks," 2022. [Online]. Available: https://www.cse.iitk.ac.in/users/sigml/lec/Slides/LSTM.pdf
[48] Z. Cui et al., "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*.
[49] R. Brueckner and B. Schulter, "Social signal classification using deep BLSTM recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2014, pp. 4823–4827.
[50] S. Bai et al., "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.
[51] C. Pelletier et al., "Temporal convolutional neural network for the classification of satellite image time series," *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 523.
[52] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[53] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Trans. Biomed. Eng.*, vol. BME-32, no. 3, pp. 230–236, Mar. 1985.
[54] N. Bayasi, H. Saleh, B. Mohammad, and M. Ismail, "65-nm ASIC implementation of QRS detector based on Pan and Tompkins algorithm," in *Proc. 10th Int. Conf. Innov. Inf. Technol.*, 2014, pp. 84–87.
[55] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[56] "RMSprop: Divide the gradient by a running average of its recent magnitude," 2014. [Online]. Available: http://www.cs.toronto.edu/tijmen/csc321/slides/lecture_slides_lec6.pdf
[57] A. Shafiee et al., "ISAAC: A convolutional neural network accelerator with in-situ analog arithmetic in crossbars," in *Proc. ACM/IEEE 43rd Annu. Int. Symp. Comput. Architecture*, 2016, pp. 14–26.

**Sumit Diware** received master's degree in VLSI design tools and technology from the Indian Institute of Technology, Delhi, New Delhi, India in 2018. He is currently working toward the Ph.D. degree with the Computer Engineering Laboratory, Delft University of Technology, Delft, Netherlands. His research interests include artificial intelligence, computation-in-memory, and neuromorphic architectures.

**Sudeshna Dash** received the bachelor's in technology degree in electronics and communication engineering from the National Institute of Technology, Tiruchirappalli, India, in 2019, and the master's degree in embedded systems with a specialization in computer architecture with the Delft University of Technology, Delft, The Netherlands, in 2021. She is currently a Design Engineer with ASML Holding N.V., Veldhoven, The Netherlands.

**Anteneh Gebregiorgis** (Member, IEEE) received the Ph.D. degree in computer science from the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2019. He is currently a Postdoctoral Researcher with the Computer Engineering Laboratory, Delft University of Technology, Delft, The Netherlands. His research interests include emerging technologies, artificial intelligence (AI), computation in-memory, neuromorphic computing, architectures for ultra-low power design, reliability analysis, and variability assessment of VLSI devices.

**Rajiv V. Joshi** (Fellow, IEEE) received the B.Tech. degree from the Indian Institute of Technology Bombay, Mumbai, India, the M.S. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, and the Dr. Eng. Sc. degree from Columbia University, New York, NY, USA. He is currently a Research Staff Member and Key Technical Lead with T. J. Watson Research Center, IBM. He has led successfully predictive failure analytic techniques for yield prediction and also the technology-driven SRAM with IBM Server Group. His statistical techniques are tailored for machine learning and AI. He developed novel memory designs which are universally accepted. He commercialized these techniques. He was the recipient of the Outstanding Technical Achievement (OTAs), three highest Corporate Patent Portfolio awards for licensing contributions, holds 60 invention plateaus and has more than 250 U.S. patents and more than 400 including international patents. He has authored or coauthored more than 200 papers. He has given more than 45 invited/keynote talks and given several Seminars. He was the recipient of the NY IP Law association Inventor of the Year Award in February 2020, prestigious IEEE Daniel Noble Award for 2018., the Best Editor Award from IEEE TVLSI journal. He was also the recipient of 2015 BMM Award. He is inducted into New Jersey Inventor Hall of Fame in August 2014. He is a Member of IBM Academy of Technology and a master inventor. He was a Distinguished Lecturer for IEEE CAS and EDS society. He is currently Distinguished Lecturer for CEDA. He is Fellow of ISQED and World Technology Network and Distinguished Alumnus of IIT Bombay.

**Christos Strydis** (Senior Member, IEEE) received the M.Sc. *(magna cum laude)* and Ph.D. degrees in computer engineering from the Delft University of Technology, Delft, The Netherlands. He holds a dual Associate-Professor position with the Neuroscience Department of the Erasmus Medical Center and with the Quantum and Computer Engineering Department, Delft University of Technology. He is also the Head of the Neurocomputing Laboratory with the Erasmus Medical Center. He has authored or coauthored work in well-known international conferences and journals. He has delivered invited talks in various venues. His research interests include brain simulations, high-performance computing, low-power embedded (implantable) systems, and functional ultrasound imaging.

**Said Hamdioui** (Senior Member, IEEE) received the M.S.E.E. and Ph.D. degrees (with Hons.) from the Delft University of Technology (TUDelft), Delft, the Netherlands. is currently the Chair Professor on dependable and emerging computer technologies, Head of the Quantum and Computer Engineering Department, and also the Head of the Computer Engineering Laboratory (CE-Lab), TUDelft, Delft, the Netherlands. He is also Co-Founder and CEO of Cognitive-IC, a start-up focusing on hardware dependability solutions. Prior to joining TUDelft as a Professor, Hamdioui spent about seven years within industry including Microprocessor Products Group with Intel Corporation, California, USA, IP and Yield Group with Philips Semiconductors R and D, Crolles, France, and DSP Design Group with Philips/ NXP Semiconductors,Nijmegen, The Netherlands. He is currently with different national and EU projects. Hamdioui received two patents, has authored or coauthored one book and contributed to other two, and coauthored more than 200 conference and journal papers. His research interests include two domains: Dependable CMOS nano-computing (including Testability, Reliability, Hardware Security) and emerging technologies and computing paradigms (including memristors for logic and storage, in-memory-computing for big-data applications). He has consulted for many companies (such as Intel, ST, Altera, Atmel, Renesas,...) in the area of memory testing and has collaborated with many industry/research partners in the field of dependable nano-computing and emerging technologies. He is strongly involved in the international community as a Member of organizing committees or a Member of the technical program committees of the leading conferences. He delivered dozens of keynote speeches, distinguished lectures, and invited presentations and tutorial at major international forums/conferences/schools and at leading semiconductor companies. Hamdioui is a an Associate Editor for IEEE TRANSACTIONS ON VLSI SYSTEMS, and he serves on the Editorial Board of *IEEE Design and Test*, *Elsevier Microelectronic Reliability Journal*, and the *Journal of Electronic Testing: Theory and Applications*. He is also a Member of AENEAS/ENIAC Scientific Committee Council (AENEAS=Association for European NanoElectronics Activities).

**Rajendra Bishnoi** received the Ph.D. degree in computer science from the Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, in 2017. He is currently an Assistant Professor with the Computer Engineering Laboratory, Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology (TU-Delft), Delft, The Netherlands. Before joining TU-Delft, he was a Research Leader for the MRAM Group in the Chair of Dependable Nano Computing, KIT for more than two years. From 2006 to 2012, he was a Design Engineer with Freescale (NXP), where he was a part of the Technical Solution Group in memory and SoC flow. His research interests include hardware AI, computation-in-memory and emerging technologies. He was the recipient of EDAA Outstanding Dissertation Award for the year 2017.