On-FPGA Spiking Neural Networks for Integrated Near-Sensor ECG Analysis

Matteo Antonio Scrugli, Paola Busia, Gianluca Leone, Paolo Meloni
DIEE, University of Cagliari
Cagliari, Italy

***Abstract*—The identification of cardiac arrhythmias is a significant issue in modern healthcare and a major application for Artificial Intelligence (AI) systems based on artificial neural networks. This research introduces a real-time arrhythmia diagnosis system that uses a Spiking Neural Network (SNN) to classify heartbeats into five types of arrhythmias from a single-lead electrocardiogram (ECG) signal. The system is implemented on a custom SNN processor running on a low-power Lattice iCE40-UltraPlus FPGA. It was tested using the MIT-BIH dataset, and achieved accuracy results that are comparable to the most advanced SNN models, reaching 98.4% accuracy. The proposed modules take advantage of the energy efficiency of SNNs to reduce the average execution time to 4.32 ms and energy consumption to 50.98 uJ per classification.**

***Index Terms*—Spiking neural networks, real-time monitoring, healthcare**

I. Introduction

In recent years, the advent of artificial intelligence and neural networks has revolutionized healthcare, offering exciting opportunities for advancements in continuous monitoring and personalized medicine. Given the burden of cardiovascular diseases, real-time monitoring of heart function represents a critical concern. In this paper, we focus on the recognition of arrhythmia, an anomaly of the rhythm of the heartbeat which can indicate more serious conditions.

The reference diagnostic tool is the electrocardiogram (ECG), whose real-time monitoring remains a significant challenge, where high accuracy standards are combined with the strict resource constraints typical of the wearable domain, opening an important area of research and development. A promising solution is represented by Spiking Neural Networks (SNNs), whose event-based processing enables high sparsity levels and results in exceptionally energy-efficient inference. Nonetheless, leveraging the advantages of event-based processing typically necessitates dedicated computational architectures.

Field Programmable Gate Arrays (FPGAs), with their highly configurable hardware design, are perfectly positioned to adapt to this computation tasks, as they can be strategically programmed to exploit the sparse firing of neurons, performing computations only on the active neurons, and bypassing the quiescent ones. The hard-wired Digital Signal Processor (DSP) slices efficiently facilitate addition, multiplication, and multiply-and-accumulate operations essential to tasks such as filters and neuron dynamics. Furthermore, flexible Block Random Access Memory (BRAM) units are instrumental in accommodating on-chip SNN models and activation data due to their customizable port widths. Lastly, programmable logic (PL) allows a broad adaptability spectrum, which is crucial given the rapid evolution of the application domain.

SNN-based systems are still a relatively young technology, and their application to the arrhythmia classification problem hasn't yet reached the remarkable accuracy levels of more complex neural networks [1]. This motivates the interest in research and development on this topic, considering the efficiency merits of this solution. In this paper, we will delve deeper into these challenges exploring the application of SNNs on low-power hardware platforms like the Lattice iCE40-UltraPlus FPGA. The main contributions can be summarized as three most relevant points:

- we present an SNN-based classification system for arrhythmia detection, reaching accuracy aligned with the state-of-the-art of SNNs;
- we explore the most effective encoding method to translate the continuous ECG signal into spike traces, assessing the advantages of exploiting delta modulation applied on the original signal, as well as on the first and second order derivatives, which results in 0.76% gain over the baseline encoding approach;
- we present an efficient implementation on the ultra-low-power Lattice FPGA, resulting in 4.32 ms inference time and 50.98 uJ energy consumption, which is over 6× lower than the state-of-the-art FPGA-based alternatives.

II. Related work

Table I summarizes the most relevant studies presenting SNN-based arrhythmia classification systems, listing the reference dataset, encoding approach, targeted device, and performance metrics.

The state-of-the-art accuracy is defined by the work of [6], where the authors proposed an ultra-energy-efficient ECG classification processor for wearable devices, presenting an SNN classifier obtained as the conversion of an Artificial Neural Network (ANN), reaching 98.6% accuracy on the MIT-BIH dataset. A remarkable 98.22% accuracy on the same dataset was also obtained in the work of [4], for the identification of the 5 most common arrhythmia classes. Both these works demonstrate exceptionally high energy efficiency, with as low as 0.3 uJ [6] and 0.75 uJ [4] per classification, evaluated on custom SNN processing chips. Other examples of efficient hardware alternatives for real-time SNN processing targeting arrhythmia detection are presented in [2], [3], [5], [7], evaluated based on the measured performance of lightweight SNN models

979-8-3503-4859-0/DATE24/© 2024 EDAA

TABLE I: A comparative summary of seminal works in ECG classification using Spiking Neural Networks.

Work	Dataset	Encoding	Classess	Accuracy in %	Device	Energy	Power
A. Amirshahi et al. [2]	MIT-BIH[1]	Poisson-distributed	4	97.9	ASIC	1.78 uJ	–
Z. Yan et al. [3]	MIT-BIH[1]	Spike rate	5	90	ASIC	–	77 mW
H. Chu et al. [4]	MIT-BIH	LC-ADC	5	98.22	ASIC	0.75 uJ	0.93 uW
Y. Liu et al. [5]	MIT-BIH	LC-ADC	4	90.5	ASIC	0.53pJ/spike	0.35 uW
R. Mao et al. [6]	MIT-BIH	Dual-Purpose Binary encoding	5	98.6	ASIC	0.3 uJ	–
J. Loh et al. [7]	MIT-BIH	Temporal coding	2	94	ASIC	–	2.2 uW
Y. Xing et al. [8]	MIT-BIH	Binary encoding	5	98.26	FPGA	346.33 uJ	250 mW
A. Rana et al. [9]	PTB ECG	Poisson-distributed	2	85	FPGA	–	176 mW
Our work	**MIT-BIH**	**Delta modulator**	**5**	**98.4**	**FPGA**	**50.98 uJ**	**11.8 mW**

[1] The reported accuracy refers to the inter-patient classification test.

reaching different accuracy levels with optimized firing rates. The models in [2], [3] deserve a dedicated mention, as the reported accuracy refers to the more challenging inter-patient classification problem, where classification is performed on the data of unseen patients.

These hardware solutions represent specialized SNN accelerators, based on custom-designed architectures, aiming at enabling the efficient exploitation of event-based sparse computation to minimize power dissipation during real-time operation. However, additional logic is required for real-time ECG monitoring, where R-peak detection is needed for appropriate signal segmentation. The test setup in [6] includes an FPGA handling data transfers, in [4] and [5] the data encoding based on level crossing analog-to-digital converter (LC-ADC) is implemented separately on a support FPGA, and neither [2] or [3] address the peak detection problem.

In this work, we take advantage of the flexibility of reconfigurable implementations based on FPGA, to accommodate peak detection, signal encoding, and SNN processing, exploiting a low-power device to reduce the energy efficiency degradation. The closest references from the literature are represented by the works of [8], [9]. In [8], the authors present an SNN integrating an attention module, reaching a competitive 98.26% accuracy on the MIT-BIH database with a 346 uJ energy consumption on the Artix-7 FPGA. We also included the contribution from [9], whose authors propose a binarized SNN, which allows for reduced operations and hardware complexity. The classification performance cannot be directly compared to the alternatives, as it was evaluated on a slightly different task, distinguishing abnormal from normal beats on the PTB Diagnostic ECG Database [10]. Nonetheless, it provides an additional reference for the power consumption of FPGA-based implementations, reporting an average of 176 mW.

To the best of our knowledge, our proposed system implementation improves the energy efficiency of the existing FPGA-based solutions. Finally, while we selected the delta modulation encoding scheme, we also introduced a novel solution that extends its application to the first and second derivatives of the ECG signal.

III. Method

In this section, we describe the details of our proposed system, presenting the reference dataset, the encoding scheme for spike generation, the selected SNN topology, and the targeted FPGA-based processor.

A. Dataset

The research presented in this paper leverages the MIT-BIH Arrhythmia Database [11], [12], an annotated dataset comprising 48 half-hour ambulatory ECG recordings collected by the BIH Arrhythmia Laboratory. The database includes a range of normal ECG and clinically significant arrhythmias, ensuring comprehensive coverage of cardiac signal dynamics.

Each recording was obtained with two-channel acquisition and digitized at a frequency of 360 Hz. Considering the impact of the electrode placement method on the signal's characteristics, we limit our analysis to single-channel acquisition with Modified Limb Lead II (MLLII), which is available for all subjects. Consistent with a majority of the literature [1], [13]–[15], our training and testing sets exclude records containing paced beats, specifically "102", "104", "107", "217". Additionally, we refer to the AAMI standard [16] as our benchmark to pinpoint the five most critical classification groups in addressing arrhythmia recognition, namely: N (non-ectopic beats), S (supra-ventricular ectopic beats), V (ventricular ectopic beats), F (fusion beat), and Q (unclassifiable beat).

B. Encoding method

As expected for SNN-based classification, the continuous signal representing the heartbeat wave needs to be encoded into spikes. We refer in this study to encoding based on delta modulation, which is commonly used in the field of signal processing and telecommunications as a technique for converting an analog signal into a digital one. Our methodology shows a distinctive element compared to existing studies in the literature, as we extended the delta modulation processing to the first derivative and the second derivative of the signal. The inclusion of the first derivative provides valuable information about the rate of change in the signal, offering insights into the rapidity of heart electrical changes. Similarly, the second derivative offers

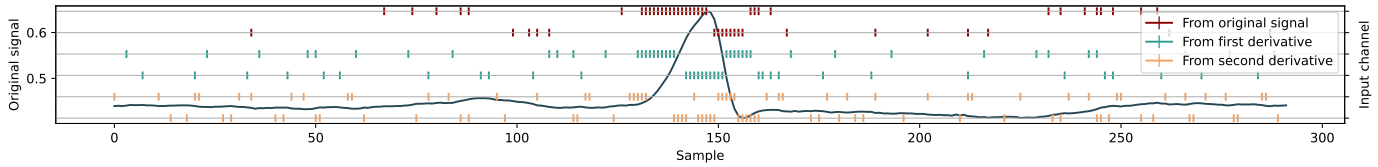

Fig. 1: Illustration of a signal window centered on the R-peak of a non-arrhythmic heartbeat with the six signals generated through delta modulation overlaid.

information about the acceleration or deceleration of these changes, further enhancing the depth of our analysis.

Delta modulation produces a pair of traces, representing positive or negative variations of the signal from a preset threshold δ. In this context, the threshold signifies either an increase or a decrease in the input signal which results in a single spike: a lower threshold will lead to spike patterns that are denser, while a higher threshold will lead to less dense spike patterns. An example of the proposed encoding scheme is provided in Figure 1, referring to a signal window centered on the QRS complex and particularly on the R-peak of a non-arrhythmic heartbeat. Over the baseline of the original heart wave, we overlay the six signals derived through delta modulation: the topmost pair traces the positive and negative differentials of the raw data, the middle pair pertains to the first-order derivative, and the bottom pair to the second-order derivative.

We normalized all signals in the range [0,1], with respect to the corresponding maximum and minimum values in the training set. Based on numerous tests, we thus selected a δ value equal to 0.003 for the original ECG signal, and one equal to 0.006 for its first and second derivative. This encoding method is described as pseudo-code in Algorithm 1.

Algorithm 1: ECG Encoding with Delta Modulation

```
Input: Normalized ECG signal
Result: Two signals of spikes, POS and NEG.
delta sample = First normalized ECG sample;
delta value;
while Normalized ECG signal do
    ECG sample;
    if ECG sample > delta sample + delta value then
        delta sample = ECG sample;
        POS.append(spike);
    else
        POS.append(no-spike);
    if ECG sample < delta sample - delta value then
        delta sample = ECG sample;
        NEG.append(spike);
    else
        NEG.append(no-spike);
```

C. SNN topology and training

The topology of our network is constructed using a sequence of Dense layers, as described in Table II. We exploited the neuron implementation provided by SLAYER [17], which works as a high-level interface for building and managing SNNs. The primary computational unit in SLAYER relies on the Loihi CUBA (Current-Based Leaky Integrate and Fire) Neuron [18]. Given the limited resources of our platform, these neurons are simplified to their LIF (Leaky Integrate and Fire).

TABLE II: Topology and parameters of the proposed SNN.

Layer	Synapse	Neuron	Delay
Dense Layer 1	6	64	True
Dense Layer 2	64	128	True
Dense Layer 3	128	64	True
Dense Layer 4	64	5	False

As shown in Equation 1 and 2, each neuron receives a series of spikes s_{in} as input. These spikes are multiplied by the synaptic weights w, contributing to the membrane voltage v within the neuron. The voltage v decays over time, influenced by a decay factor α.

$$v'(t) = \alpha \cdot v(t) + \sum w \cdot s_{in}(t) \tag{1}$$

$$v(t+1) = \begin{cases} v'(t), & \text{if } v'(t) < \theta \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

As shown in Equation 3, the neuron emits an output spike $s_{out}(t)$ when its voltage exceeds a certain threshold θ. The output spike can be represented as:

$$s_{out}(t+1) = \begin{cases} 1, & \text{if } v'(t) > \theta \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

The SLAYER toolbox is particularly useful due to its support for defining custom spiking neurons and synaptic behaviors. We exploited it in conjunction with the LAVA framework [19], [20], providing efficient GPU-based computations for the training and simulation of the SNN.

The training process was executed on the Google Colab platform, leveraging the computational power of T4 GPUs. We exploited the Adam optimizer with a learning rate varying between 0.001 and 0.00001, and batch sizes set to 32. The *slayer.loss* class is used for loss calculation, specifically opting for the *SpikeRate* based loss calculation. It serves to compute the error, setting a true rate target of 0.2 and a false rate of 0.03. To prevent overfitting, we exploited the *early stopping* method, with a patience parameter set to 20 epochs. We referred to the intra-patient testing approach, thus randomly splitting the

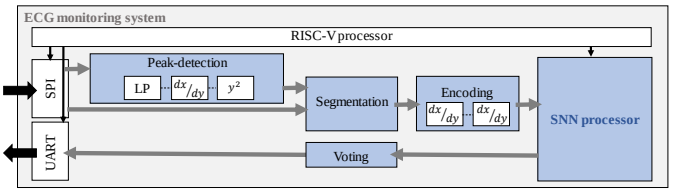

Fig. 2: Overview of the ECG monitoring system.

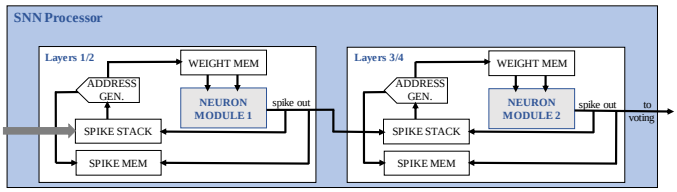

Fig. 3: Architecture of the SNN processor.

dataset into 70% for training, 10% for validation, and 20% for testing.

Finally, the MIT-BIH dataset represents a highly unbalanced resource, where roughly 89.5% of the heartbeats belong to the N class, only 2.8% to the S class, around 7% to the V class, 0.8% to the F class, and only a negligible percentage to the Q class. To alleviate this issue, we exploited minority classes augmentation on the training set, considering shifted windows around the center of the heartbeats.

D. System Overview

We provide in Figure 2 an overview of the implemented system. General supervision tasks are handled by a RISC-V processor, specifically utilizing the SERV [21] core. Its duties include managing the input-output peripherals, overseeing the ECG signal acquisition through the SPI interface based on the desired sampling frequency, and transferring the output classification through the UART interface. It also handles the system initialization.

After signal acquisition, the dataflow includes two branches. The first one, the *peak-detection* block, is an auxiliary module, exploited for locating the position of the R-peak to properly frame the heartbeat when performing signal segmentation. This block implements a simplified version of Pan-Tompkins algorithm [22], through the cascade of a 4th-order IIR low-pass filter with 11Hz cut-off frequency, a derivative block and a squared block. The second branch propagates the acquired signal towards the segmentation block, which also receives the peak detection output. This framing block selects a window centered on the R-peak of the QRS complex in the heartbeat, with a width of 292 samples.

In order to be processed for classification, the signal needs to be translated into a train of spikes: the segmented windows thus go through the *encoding* stage, which is implemented as two derivative filters, based on the scheme described in Section IV-A. At this point, the spike traces are provided as inputs to the SNN processor, which is described in detail in Section III-E. The output of the SNN finally passes through a *voting* stage, which produces the classification to be propagated in output through the UART interface.

E. SNN processor

Figure 3 describes the structure of the SNN processor used in the system, inspired from the work in [23]. It receives the input spikes from the encoding module. The processor integrates two *neuron* modules capable of accumulating four 8-bit synaptic weights per cycle each, to compute the synaptic current. The two neuron modules are also in charge of computing the LIF dynamics to identify output spikes.

Each neuron module is connected to a weight memory, filled at the boot with the networks weights derived from training, and to a spike memory, storing the results of previous inferences for all the neurons in the SNN. An *address generator* computes address to such memories to reuse the hardware cores to iterate the processing of all the neurons in the SNN layers. More specifically, each module is in charge of processing two layers of the SNN described in Table II.

A *spike stack* feeds the address generator with an indication of the location of active spikes inside the spike memory, in order to enable skipping of inactive synapses to take profit from the sparsity of events. The output spikes are passed in output to the voting module to be counted for obtaining classification.

IV. Experimental Results

In this section, we present the experimental results concerning the performance assessment of the proposed ECG monitoring system, in terms of classification accuracy and on-hardware inference efficiency.

A. Encoding Assessment

We started from the evaluation of the effectiveness of our proposed encoding scheme. We examined three distinct scenarios, progressively increasing the set of signals available for classification: considering delta-modulation applied only to the raw signal; including its first-order derivative; including also the second-order derivative. We thus performed three training trials, including up to six traces of spikes as input to the SNN.

Considering as a baseline the first scenario, limiting the input to the raw signal, the inclusion of the first-order derivative resulted in an accuracy gain of 0.6% points, which was further increased to 0.76% points by the additional information provided by the second-order derivative. These numbers suggest that leveraging the raw signal in conjunction with its derivatives, processed through the delta modulation algorithm, can supply SNNs with highly informative inputs, thereby significantly enhancing their performance. As can be noticed from Figure 1, each pair of resulting traces offers an individualized view of the intricate dynamics underlying the raw signal and its derivatives. We thus considered the six-input model as the best solution for the results reported in the following.

B. Classification performance

Targeting memory footprint reduction and increased efficiency on the considered hardware system, the selected model

TABLE III: Confusion matrix reporting 8-bit classification performance on the test set.

True Labels \ Predicted Labels	N	S	V	F	Q
N	17851	91	34	20	0
S	82	386	5	0	0
V	35	7	1051	4	0
F	26	3	7	113	0
Q	3	0	1	0	0

was quantized reducing the precision of the neurons' parameters to 8 bits. The evaluation of the classification performance on the test set resulted in 98.4% accuracy. We include for completeness in Table III the corresponding confusion matrix, providing an assessment of the classification performance on the different arrhythmia classes.

C. Sparsity

SNNs are known to inherently enforce sparsity because, at any given moment, only a minor fraction of neurons are actively firing or communicating. We calculated the overall sparsity across the SNN layers by Equation 4.

$$Sparsity = \left(1 - \frac{\sum_{i=1}^{L}\sum_{j=1}^{N_i} n_{ij}}{\sum_{i=1}^{L} N_i}\right) \times 100 \qquad (4)$$

Where:

- L is the total number of layers in the network,
- N_i is the number of output neurons in the i-th layer,
- n_{ij} indicates whether the j-th neuron in the i-th layer is active (1 if active, 0 otherwise).

The equation calculates the proportion of non-active spikes as a percentage of the maximum possible number of spikes in the network. We obtained a sparsity value of 92%, resulting from having on average only 8% of active potential spikes. Figure 4 provides a detailed layer-by-layer breakdown of the spike distribution across the layers and the influence of the network's architecture on sparsity, averaged over the test set.

At this point, it is crucial to consider to which extent the sparsity resulting from the model architecture can be exploited on the target hardware, and translated into performance efficiency. A significant constraint deriving from the parallelism enforced on the platform requires the aggregation of spikes into groups of four. Thus, while the reported 92% sparsity offers valuable insights into the sparse nature of computations in SNNs, the effective sparsity available during inference on the target platform could be lower, due to the spike clustering. This new factor, which we will call *sparsity-hw*, is calculated from Equation 5.

$$Sparsity\text{-}hw = \left(1 - \frac{\sum_{i=1}^{L}\sum_{j=1}^{G_i} g_{ij}}{\sum_{i=1}^{L} G_i}\right) \times 100 \qquad (5)$$

Where:

- G_i is the number of spike groups in the i-th layer,
- g_{ij} indicates whether there's at least one active spike in the j-th group in the i-th layer.

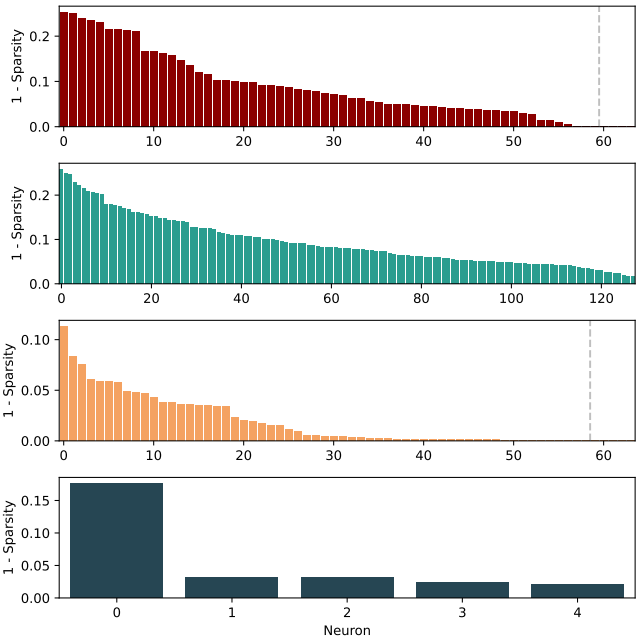

Fig. 4: Sparsity distributions for each neuron in layers (from top to bottom) 1, 2, 3, and 4. The semi-transparent vertical line indicates the point where neurons with zero activity begin.

In our case, we obtained a sparsity-hw value of 85%.

D. Power consumption

We finally evaluated the energy efficiency of the proposed FPGA implementation for our selected SNN model. To account for the advantages of variable sparsity, we used a mathematical approach for the evaluation of the average execution time through the test set.

We calculated a number of expected execution cycles equal to approximately 97,205 cycles, considering the parallel resources available on our target accelerator, based on Equation 6.

$$execution\ cycles = \frac{ops \cdot (1 - sparsity\text{-}hw)}{modules \cdot parallel\ spikes} \qquad (6)$$

Specifically, in order to obtain the number of cycles required for a single inference, we first consider the maximum number of operations provided by the selected network topology, which in this case is equal to 5,083,136 *ops*. Subsequently, we included two multiplicative factors to account for hardware-level parallelism, integrating two neuron modules that work in a pipeline and process four spikes simultaneously. Finally, we accounted for the performance improvement deriving from the sparsity-hw. We obtained a classification time of 4.32 ms at 22.5 MHz working frequency. To evaluate power metrics, we exploited a Digilent Analog Discovery 2 oscilloscope and three $1.0 \pm 0.01\ \Omega$ shunt resistors. The resistors were connected to the three different power inputs of the Lattice iCE40UP5k FPGA, labeled as Vcore, VCCIO0&1, and VCCIO2. Vcore provides 1 V to the internal components of the FPGA, while the VCCIO0&1 and VCCIO2 circuits supply 3.3 V to the I/O pins. We assessed the average power consumption to be 11.8 mW during

classification, data acquisition, and processing. Considering the previously calculated network execution duration, the energy consumption for classification inference is equal to 50.98 uJ.

E. Discussion

Our proposed monitoring system reaches a classification accuracy comparable with the best SNN model in the literature [6], and higher than all the other alternatives examined in Table I. The high sparsity levels observed enable remarkable classification time and energy savings, which, combined with the efficiency of the proposed hardware design, result in the lowest energy consumption among the FPGA-based solutions [8], [9]. As can be expected, the energy efficiency of ASIC-based alternatives, showing as low as 0.3 uJ per inference [6], remains unmatched. Nonetheless, the proposed reconfigurable solution allows to allocate all the data preparation operations in a single device, which can be effectively exploited for standalone real-time ECG monitoring without the need for any additional computing unit.

V. Conclusion

In this work, we presented a low-power ECG monitoring system for the real-time detection of arrhythmia, exploiting SNN-based classification to leverage the efficiency of event-based computation. We explored the most effective signal encoding approach, to translate the heartbeat wave into a train of spikes, showing the advantages, up to a 0.76% accuracy gain, of exploiting delta modulation and including the first- and second-order derivatives of the signal, providing different levels of information about the rate of change of the signal. The proposed model reaches 98.4% classification accuracy in the recognition of the 5 most relevant arrhythmia classes of the MIT-BIH dataset, a result competitive with the alternatives in the literature. We demonstrated the high sparsity levels enabled by the proposed classification scheme, with roughly only 15% active neurons on each inference run. Finally, we presented a low-power FPGA-based implementation, exploiting a custom design for SNN acceleration. The measurements of on-hardware performance showed as low as 4.32 ms classification time and 50.98 uJ energy consumption.

Acknowledgment

This work was supported by Key Digital Technologies Joint Undertaking (KDT JU) in EdgeAI "Edge AI Technologies for Optimised Performance Embedded Processing" project, grant agreement No 101097300.

References

[1] G. Yan, S. Liang, Y. Zhang, and F. Liu, "Fusing transformer model with temporal features for ecg heartbeat classification," in *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2019, pp. 898–905.

[2] A. Amirshahi and M. 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*, vol. 13, no. 6, pp. 1483–1493, 2019.

[3] Z. Yan, J. Zhou, and W.-F. Wong, "Energy efficient ecg classification with spiking neural network," *Biomedical Signal Processing and Control*, vol. 63, p. 102170, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809420303098

[4] H. Chu, Y. Yan, L. Gan, H. Jia, L. Qian, Y. Huan, L. Zheng, and Z. Zou, "A neuromorphic processing system with spike-driven snn processor for wearable ecg classification," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 4, pp. 511–523, 2022.

[5] Y. Liu, Z. Wang, W. He, L. Shen, Y. Zhang, P. Chen, M. Wu, H. Zhang, P. Zhou, J. Liu, G. Sun, J. Ru, L. Ye, and R. Huang, "An 82nw 0.53pj/sop clock-free spiking neural network with 40μs latency for alot wake-up functions using ultimate-event-driven bionic architecture and computing-in-memory technique," in *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, vol. 65, 2022, pp. 372–374.

[6] R. Mao, S. Li, Z. Zhang, Z. Xia, J. Xiao, Z. Zhu, J. Liu, W. Shan, L. Chang, and J. Zhou, "An ultra-energy-efficient and high accuracy ecg classification processor with snn inference assisted by on-chip ann learning," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 16, no. 5, pp. 832–841, 2022.

[7] J. Loh and T. Gemmeke, "Lossless sparse temporal coding for snn-based classification of time-continuous signals," in *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023, pp. 1–6.

[8] Y. Xing, L. Zhang, Z. Hou, X. Li, Y. Shi, Y. Yuan, F. Zhang, S. Liang, Z. Li, and L. Yan, "Accurate ecg classification based on spiking neural network and attentional mechanism for real-time implementation on personal portable devices," *Electronics*, vol. 11, no. 12, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/12/1889

[9] A. Rana and K. Kim, "A novel spiking neural network for ecg signal classification," *JOURNAL OF SENSOR SCIENCE AND TECHNOLOGY*, vol. 30, pp. 20–24, 01 2021.

[10] R. Bousseljot, K. D., and S. A., "Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet," *Biomedizinische Technik / Biomedical Engineering*, pp. 317–318, 1995.

[11] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[12] A. Goldberger, L. Amaral, L. Glass, J. Hausdorff, P. C. Ivanov, R. Mark *et al.*, "Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals." *Circulation*, vol. 101, p. e215–e220, 2000.

[13] R. Hu, J. Chen, and L. Zhou, "A transformer-based deep neural network for arrhythmia detection using continuous ecg signals," *Computers in Biology and Medicine*, vol. 144, p. 105325, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482522001172

[14] M. M. Farag, "A tiny matched filter-based cnn for inter-patient ecg classification and arrhythmia detection at the edge," *Sensors*, vol. 23, no. 3, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/3/1365

[15] T. Wang, C. Lu, Y. Sun, M. Yang, C. Liu, and C. Ou, "Automatic ecg classification using continuous wavelet transform and convolutional neural network," *Entropy*, vol. 23, no. 1, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/1/119

[16] A. for the Advancement of Medical Instrumentation and A. N. S. Institute, *Testing and Reporting Performance Results of Cardiac Rhythm and ST-segment Measurement Algorithms*, ser. ANSI/AAMI. The Association, 1999. [Online]. Available: https://books.google.it/books?id=gzPdtgAACAAJ

[17] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 1419–1428.

[18] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, P. Joshi, A. Lines, A. Wild, H. Wang, and D. Mathaikutty, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. PP, pp. 1–1, 01 2018.

[19] "LAVA-NC," 2023, accessed: 2023-1-8. [Online]. Available: https://lava-nc.org/index.html

[20] "Lava-DL SLAYER," 2023, accessed: 2023-1-8. [Online]. Available: https://lava-nc.org/lava-lib-di/slayer/slayer.html

[21] "SERV," 2023, accessed: 2023-15-12. [Online]. Available: https://github.com/olofk/serv

[22] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[23] G. Leone, L. Raffo, and P. Meloni, "On-fpga spiking neural networks for end-to-end neural decoding," *IEEE Access*, vol. 11, pp. 41 387–41 399, 2023.