Weightless Neural Networks on Flexible Substrates: A Novel Approach to Wearable Machine Learning

Igor D. S. Miranda, Velu Pillai, *Member, IEEE*, Tejas Musale, Mugdha Jadhao, Paulo C. R. Souza Neto, Zachary Susskind, *Member, IEEE*, Alan Bacellar, Mael Lhostis, Priscila M. V. Lima, Diego L. C. Dutra, Eugene B. John, *Life Senior Member, IEEE*, Mauricio Breternitz Jr., *Life Senior Member, IEEE*, Felipe M. G. França, *Senior Member, IEEE*, Emre Ozer, *Member, IEEE*, and Lizy K. John, *Fellow, IEEE*

***Abstract*—In this article, we present a novel approach that seamlessly integrates machine learning (ML) algorithms into wearable technology through the use of weightless neural networks (WNNs) and flexible integrated circuits (FlexICs). Our methodology employs combinational intelligent networks (COIN) for edge inference on resource-constrained devices, highlighting the advantages of WNNs in terms of power efficiency and minimal hardware requirements. We propose an automated design flow for implementing COIN as FlexICs aimed at developing scalable, cost-effective, and environmentally sustainable wearable monitoring solutions. As a proof-of-concept demonstrator, an arrhythmia detection FlexIC was fabricated using COIN to meet the stringent requirements of medium-complexity wearable applications, offering a promising path toward personalized and accessible healthcare solutions.**

***Index Terms*—Automated design flow, edge computing, flexible integrated circuits (FlexICs), wearable devices, weightless neural networks (WNNs).**

I. Introduction

IN THE rapidly evolving domain of wearable technology, seamlessly integrating sophisticated machine learning (ML) algorithms into hardware emerges as a formidable challenge. This is especially pronounced within the healthcare sector, where the promise of wearable devices to deliver revolutionary, personalized, and widely accessible health solutions meets the reality of stringent resource constraints. These limitations—spanning area, cost, and notably, power consumption—severely restrict the viability of numerous wearable applications.

Flexible integrated circuits (FlexICs) [1] emerge as a disruptive alternative to silicon-based ICs for healthcare devices. They are fabricated using thin-film transistors (TFTs) on a flexible polyimide substrate. FlexICs are ultralow-cost and physically flexible, allowing conformability. The current FlexIC technology offers resistive n-type logic, where a resistive pull-up is coupled with a n-type TFT because of the lack of a p-type device. This means that digital FlexICs are clocked at kHz rates, and very large-scale integration is challenging due to high static power consumption. Hence, implementing complex modern ML inference models as FlexICs is constrained by reduced clock frequencies and higher power consumption. Suitable ML models must use computationally simple operations and fewer hardware resources to be implemented as ML hardware accelerators in FlexICs. As FlexICs are very quick to fabricate (weeks instead of months for silicon process), our expedited design flow enables agile creation of intelligent wearable devices.

Pioneering studies have demonstrated the feasibility of implementing ML algorithms on FlexICs for specific applications. Ozer et al. [2] present the concept of a framework for developing ML hardware customized for applications fabricated on flexible substrates. The prior studies in [3], [4], and [5] show the successful development of odor classifiers implemented and fabricated as FlexICs. Ozer et al. [6] develop a FlexIC detecting atrial fibrillation events, but the FlexIC is not based on an ML model. The method proposed in [7] is an evolutionary algorithm framework that generates tiny circuits directly from tabular data, which was also used to implement and fabricate FlexICs. In all of these studies, classification circuits fabricated as FlexICs are dedicated circuits for specific applications. Except for the work in [7], none of them stand out as a general-purpose framework. Additionally, all of them only handled data with a small number of features, which raises questions regarding their adequacy for the medium-complexity data typically encountered in wearable device applications.

To address these challenges in implementing ML algorithms on FlexICs, we advocate for the adoption of combinational intelligent networks (COIN) [8]. This ML algo-

Received 19 April 2025; revised 12 October 2025; accepted 2 November 2025. Date of publication 25 November 2025; date of current version 23 January 2026. This work was supported in part by Semiconductor Research Corporation (SRC) Task 3148.001; in part by the National Science Foundation (NSF) under Grant 2326894 and Grant 2425655; in part by the NVIDIA Applied Research Accelerator Program Grant; in part by FCT/MCTES through national funds, and when applicable, co-funded by EU funds under Project UIDB 50008/2020, Project UIDB/04466/2020, and Project UIDP/04466/2020; and in part by the Next Generation EU through PRR Project Route 25 under Grant C645463824-00000063. *(Corresponding author: Igor D. S. Miranda.)*

Igor D. S. Miranda and Paulo C. R. Souza Neto are with the Division of Electrical and Computer Engineering, Federal University of Recôncavo da Bahia, Cruz das Almas 44380-000, Brazil (e-mail: igordantas@ufrb.edu.br).

Velu Pillai and Eugene B. John are with the Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249 USA.

Tejas Musale, Mugdha Jadhao, Zachary Susskind, and Lizy K. John are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA.

Alan Bacellar, Priscila M. V. Lima, and Diego L. C. Dutra are with the Federal University of Rio de Janeiro, Rio de Janeiro 21941-914, Brazil.

Mael Lhostis and Emre Ozer are with Pragmatic Semiconductor, CB4 0WH Cambridge, U.K.

Mauricio Breternitz is with the ISTAR Laboratory, ISCTE—Instituto Universitário de Lisboa, 1649-026 Lisbon, Portugal.

Felipe M. G. França is with the Instituto de Telecomunicações, Universidade do Porto, 4200-465 Porto, Portugal.

Digital Object Identifier 10.1109/TVLSI.2025.3630337

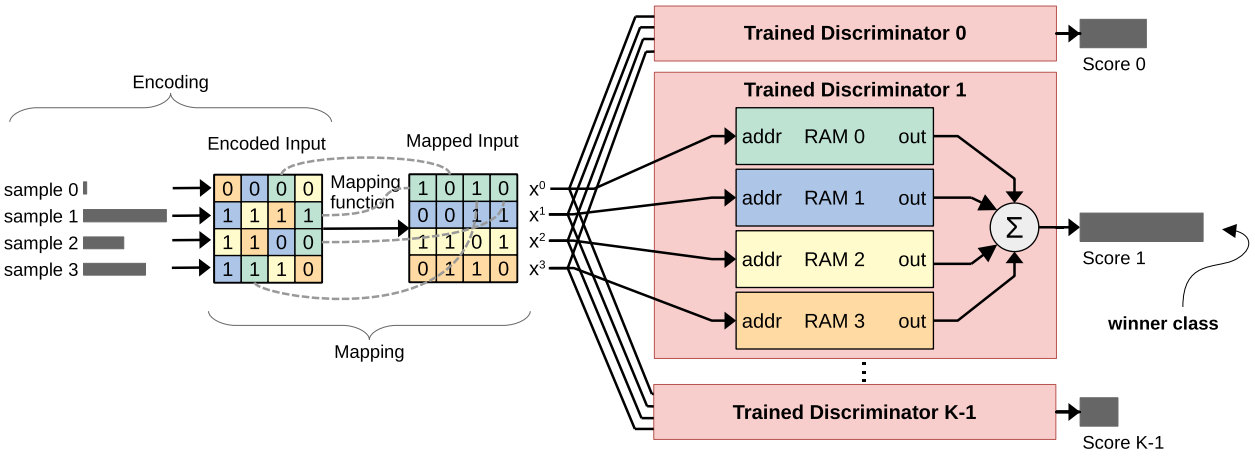

Fig. 1. Overview of a K-class WiSARD classifier. Input data, preprocessed using thermometer encoding and randomly defined mapping, are applied to multiple discriminators (1 per class). Each discriminator performs lookups using RAM nodes and generates a score. The highest score discriminator indicates the winner class.

rithm is specifically designed for efficient edge inference on resource-constrained platforms, such as field-programmable gate arrays (FPGAs) and application-specific integrated circuits (ASICs). It performs pattern classification directly on-device using a lightweight digital architecture, enabling real-time operation under strict area and power constraints. Building upon the LogicWiSARD [9] model, COIN is distinguished by its minimal power needs, small footprint, and high efficiency—traits that align perfectly with the constraints of FlexIC technology.

This work introduces a novel automated design flow for implementing COIN as FlexICs, aimed at realizing scalable, cost-effective, and environmentally sustainable health monitoring solutions. Furthermore, we present a proof-of-concept heart arrhythmia detector, fabricated using our proposed methodology, to demonstrate the viability and effectiveness of this integrated approach in real-world healthcare monitoring.

The main contributions of this work are as follows.

1) We propose a novel automated design flow for implementing COIN as FlexICs, with support for training, model selection, and register transfer level (RTL) generation.
2) We demonstrate that COIN's fully combinational architecture aligns naturally with the physical and electrical constraints of FlexIC fabrication.
3) We introduce a gate-aware training and model selection strategy that enables compact and efficient hardware implementations.
4) We provide a fully open-source implementation of our methodology to promote reproducibility and further research in logic-based ML for wearable devices.

II. Background

A. WNNs and LogicWiSARD

Weightless neural networks (WNNs [10]) are ML algorithms that use several lookup tables (LUTs) to implement pattern detectors. A WNN detector, often called a *discriminator*, generates a confidence score according to the number of LUT positions accessed and their values. A multiclass classifier uses multiple discriminator, as depicted in Fig. 1, and is known as Wilkie, Stonham, and Aleksander recognition device (WiSARD) [11].

The input for a WNN model must be a set of bits that are shuffled and split into groups according to a mapping function. Each group of n bits is used as an n-tuple address for an associated LUT, also called RAM node, in each discriminator. If a task has real-valued inputs, an encoding technique called thermometer [12] is usually adopted to provide low-dimension input data to WNNs. Fig. 1 also illustrates a typical data preprocessing for WNN inference.

Recent WNN works have demonstrated how to reduce the required memory for a WiSARD model without hurting performance by binarizing memories' content during training [9], [13]. Relying on logic functions for processing, LogicWiSARD technique exemplifies the utilization of WNNs for efficient computation, eliminating the need for the extensive memory and computational resources typical of traditional neural networks.

The LogicWiSARD model is characterized by its minterms, denoted as m_n^r, and outputs of the truth table, $\tau_{n,k}^r$, with $r = 0, \ldots, R-1$ representing the index of the RAM, $n = 0, \ldots, N^r-1$ signifying the index of the minterm within a RAM that contains N^r minterms, and k indicating the index of the class. An illustration of a LogicWiSARD model after training is shown in Fig. 2. After training, address frequencies in each RAM node are binarized using a validation-driven threshold that retains only the most discriminative patterns. The remaining active addresses are then converted into Boolean minterms, defining the final decision logic used by COIN.

Upon encoding and assigning, an input x generates R addresses x^r, which proceed to the classification stage. The LogicWiSARD model, designed to handle K classes, computes the predicted class as

$$\widehat{C} = \text{argmax}(\theta_k) \tag{1}$$

where $k = 0, \ldots, K-1$

$$\theta_k = \sum_{r=0}^{R-1} \sum_{n=0}^{N^r-1} \tau_{n,k}^r q(m_n^r, x^r), \tag{2}$$

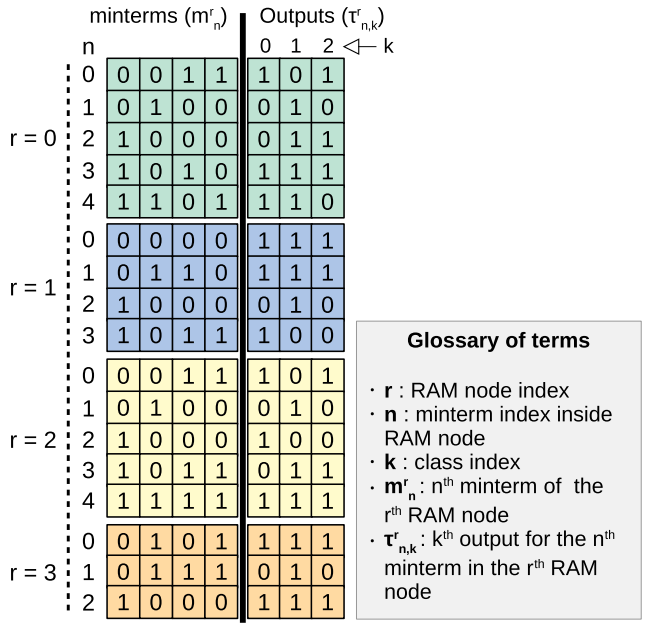

Fig. 2. Depiction of a LogicWiSARD model that is trained, featuring four RAM nodes and three classes of output as per [9]. The count of minterms per RAM node varies from 3 to 5 in this instance.

and

$$q(a,b) = \text{AND}_{\text{unary}}(\text{XNOR}_{\text{bitwise}}(a,b)) \tag{3}$$

where θ_k is the score for the discriminator corresponding to class k. The function $q(a,b)$ is designed to yield 1 if its inputs match and 0 otherwise.

In (2), the inner summation represents the search for the input address x^r among the minterms in a RAM node, which is actually implemented as a multiplexer. Whenever a minterm is found, the output value $\tau^r_{n,k}$ is added to the total score computation.

B. Combinational Intelligent Networks

The development and application of COIN represent an innovative step forward in the realm of WNNs [8]. Originating from the LogicWiSARD model, COIN enhances the practicality of WNNs by offering a solution that balances computational simplicity with improved performance.

The transition from LogicWiSARD to COIN involves a distinctive training regimen that bridges the gap between WNNs and BNNs [14], the previous state-of-the-art binary neural networks. This process starts with translating a LogicWiSARD model into a BNN format, subsequently applying backpropagation for training. This intermediary BNN serves as the scaffold for COIN, which, after a final transformation, emerges as a refined WNN. Note that this transformation is exact and does not introduce quantization, ensuring that the model's accuracy is preserved in hardware implementation.

Other than the training strategy, COIN holds some differences to LogicWiSARD regarding the encoding and the network representation. COIN uses the reverse ripple thermometer (RRT) encoding, which is particularly effective for input samples that exhibit an approximately exponential distribution. Another difference is that the values, which represent the model's knowledge, are $\{-1,+1\}$ instead of $\{0,1\}$ as in $\tau^r_{n,k}$. Despite these distinctions, the COIN model bears resemblance to the equation shown in (2). It is succinctly expressed by the following equation:

$$\widehat{\theta' k} = \sum r = 0^{R-1} \sum_{n=0}^{N^r-1} (\omega' k, lq(mn^r, x^r) - \sigma_k^+) \tag{4}$$

where $\omega_{k,l}$ represents the trainable parameters of the model, and

$$\sigma_k^+ = \frac{\sigma_k}{\sqrt{\frac{1.5}{N+K}}} \tag{5}$$

denotes our batch normalization technique, where σ_k is the batch normalization strategy introduced by FINN [14].

Diverging from the memory-intensive architectures of conventional neural networks, COIN, following in the footsteps of LogicWiSARD, adopts a logic-based framework that obviates the necessity for memory. This paradigm shift facilitates a more compact and energy-efficient hardware design, which seamlessly integrates into technologies with stringent constraints, such as FlexICs.

The hardware architecture of the generated HDL is shown in Fig. 3. COIN takes one tuple every clock, which is sent to the minterm group associated with its index. Since the model stores $\{-1,+1\}$, up/down counters are used to compute the score of each class discriminator. After all tuples are presented, the discriminator with the highest score defines the predicted class.

As COIN models are entirely represented with minterm equations, as described in Section II-A, its hardware is mainly combinational. The sequential part is proportionally tiny and depends only on the number of classes. Also, note that its hardware does not contain any arithmetic circuitry. With improvements in accuracy, resource utilization, and energy efficiency, COIN is an alternative for edge inference using FPGAs or ASICs.

In terms of hardware footprint, COIN-based classifiers are directly influenced by several factors related to the complexity of the learned decision boundaries. These include the number of output classes, the structure of the mapping function, the encoding scheme (e.g., thermometer encoding), and hyperparameters, such as address size and thermometer resolution. Collectively, these factors determine the number of minterms required to represent the classifier, which in turn correlates directly with the gate count in the synthesized circuit.

C. FlexIC Technology

The FlexIC technology is provided by Pragmatic semiconductor [15]. FlexICs are fabricated in Pragmatic's low-cost, low-carbon footprint [16], and fast turnaround (i.e., tape-out to fab-out) fabs enabling the development of ultralow-cost and conformable ICs. These qualities make FlexICs particularly appealing for applications across healthcare, wearable technologies, fast moving consumer goods, and the broader Internet of Things (IoT) ecosystem, where flexibility, form factor, and cost are the critical parameters.

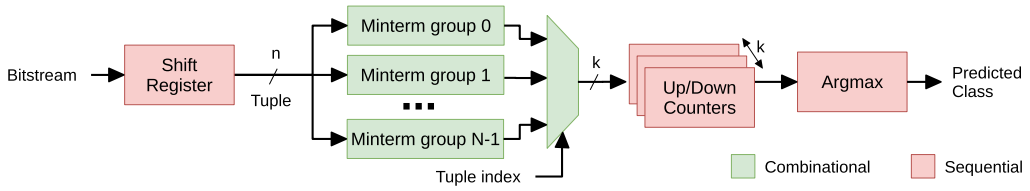

Fig. 3. Hardware architecture for COIN. Minterm groups may be implemented as logic gates or LUTs. Each of the k up/down counters has an initialization value to implement the batch normalization constant σ_k^+.

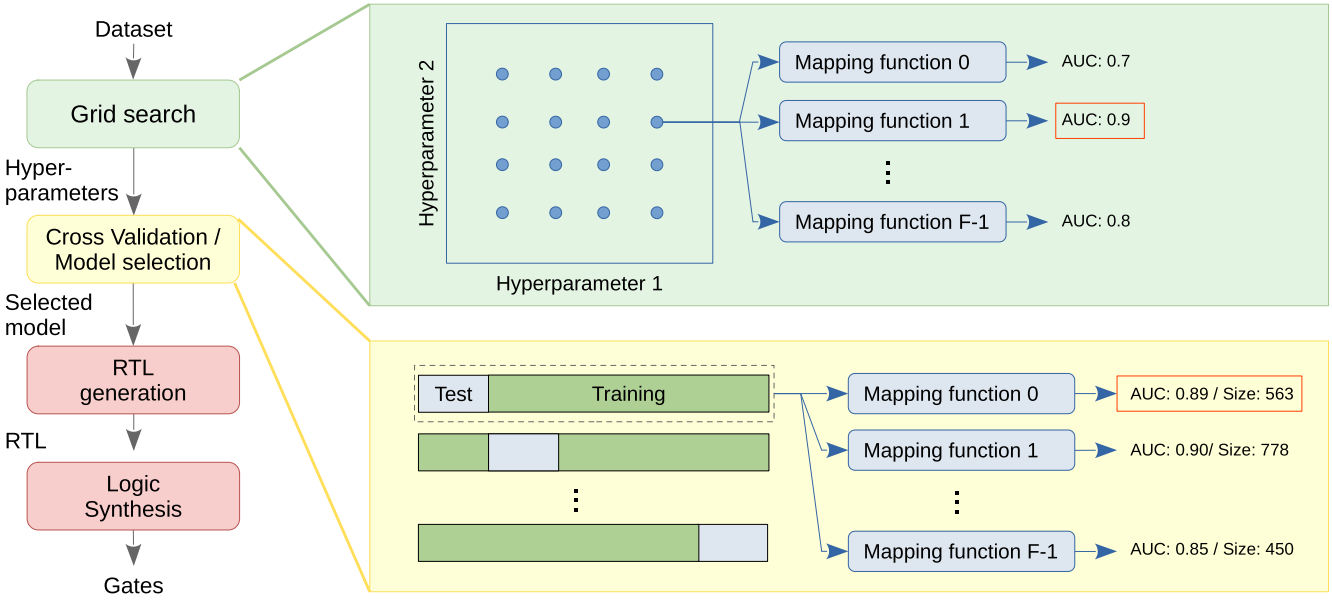

Fig. 4. Flowchart illustrating the four-stage design process for the proposed methodology: starting with a grid search that evaluates various hyperparameter and mapping function combinations, followed by cross-validation/model selection using a Monte Carlo approach to achieve optimal AUC and minimal minterms. The process culminates in RTL generation and logic synthesis, transforming the COIN model into a physical design optimized for flexible electronics.

The underlying technology for FlexICs employs TFT technology that is based on a metal–oxide material called indium–gallium–zinc oxide (IGZO) to fabricate circuits on flexible substrates, such as polyimide. Such substrates can endure bending to a minimal radius of curvature of 5 mm, ensuring the integrity of the circuitry under physical stress. The IGZO TFTs have channel lengths of 0.6 μm and can operate at a nominal supply voltage of 3 V. The combination of physical flexibility, ultralow cost, and fast turnaround in the production of FlexICs will revolutionize electronic devices by integrating them into unconventional, flexible form factors without compromising functionality.

III. Automatic Design Flow for COIN-FlexIC Implementations

The implementation of COIN as FlexICs necessitates an innovative design flow that minimizes human intervention while accommodating moderate complexities. This section elucidates an automated design methodology comprising four pivotal steps: Grid search, cross-validation/model selection, RTL generation, and logic synthesis. The design flow is depicted in Fig. 4. The essence of this process lies in its capacity to streamline the design cycle, enhancing both the efficiency and scalability of wearable health monitoring technologies and making the most of the rapid fabrication advantages of FlexICs to broaden the range of applicability of the proposed approach.

A distinctive aspect of the COIN training process is the critical role played by the mapping function. This function, which shuffles the input bits of the WNN models, is randomly defined during training and remains constant during inference. Our empirical analyses reveal that the choice of mapping function significantly influences both the performance and size of the model, albeit without a discernible pattern. To navigate this ambiguity, our methodology entails evaluating various randomly defined mapping functions during the grid search phase, selecting the one that optimizes performance metrics.

At the core of COIN model representation is the concept of minterms, which later facilitate the conversion to gate-level representations. In pursuit of technology agnosticism, these minterms are encoded as case statements in the RTL code, offering a predictive measure of the final model's size based on the linear relationship between the number of minterms and the gates mapped onto ASIC or FPGA technologies. This representation not only ensures compatibility across different hardware platforms but also aids in the estimation of resource requirements.

The design flow can be well understood through its four stages, beginning with a grid search that parallels traditional methods plus an evaluation of random mapping functions alongside hyperparameter combinations. The subse-

quent cross-validation/model selection stage employs a Monte Carlo cross-validation approach, with a novel twist: for each iteration, different mapping functions are assessed and model selection is based on achieving an average area under the curve (AUC) that surpasses 99% of the cohort, with a preference for models exhibiting minimal minterms. RTL generation follows, automating the creation of a general-purpose RTL code that encapsulates the intricacies of the COIN model within a technology-independent framework. Finally, logic synthesis transforms this RTL code into a physical design using FlexIC standard cells, tailored to meet the stringent requirements of flexible electronics.

The proposed automated design flow benefits wearable technology development by streamlining the design process with reduced human intervention. This method not only speeds up the design cycle but also ensures a size-conscious model selection, essential in FlexIC technology where minimizing device footprint and power consumption is crucial. Moreover, this approach facilitates the efficient development of medium-complexity classifiers, offering remarkable scalability and adaptability. The effectiveness and broader applications of this design flow will be further detailed in our work.

While our methodology automates the flow from model training to technology-agnostic RTL generation, the physical implementation (place and route, layout, and GDSII generation) follows the standard digital design flow using commercial EDA tools. These back-end stages currently require manual setup and design supervision, which motivates future efforts toward a fully automated physical layout pipeline.

IV. Design Example: Arrhythmia Detection

This section demonstrates the application of our methodology to develop and evaluate a COIN-based arrhythmia detector implemented as a FlexIC device.

A. Dataset

We adopted the MIT-BIH Arrhythmia dataset [17] for our proof-of-concept implementation due to its widespread acceptance and detailed electrocardiogram (ECG) recordings. While its rich variety of arrhythmic patterns provides a starting point for developing wearable monitors, the dataset's limited number of patients underscores the challenge of model generalization. However, this limitation, in turn, propels the utilization of the COIN methodology, which is well-suited to generalize effectively in these scenarios.

The MIT-BIH dataset provides an ideal scenario for applying WNNs within our COIN methodology. By reducing the dataset from five to two classes—normal and abnormal—we simplify the model significantly without sacrificing diagnostic accuracy, which is particularly beneficial in the context of wearable devices and FlexICs. In practical terms, this example device could be invaluable for initial triage.

Although the MIT-BIH dataset contains a limited number of patients, this reflects a realistic scenario for personalized healthcare, where small datasets are often used to develop wearable devices tailored to specific populations. The low-cost and rapid fabrication cycle of FlexICs makes it feasible to produce application-specific models for targeted groups with limited data availability.

B. Preprocessing

The data preprocessing pipeline for our arrhythmia classification system commences with the application of the Pan–Tompkins algorithm (PTA) bandpass filter [18]. This algorithm, renowned for its efficacy in discerning QRS complexes (a characteristic waveform segment of ECG signals representing ventricular depolarization) within ECG signals, serves as a foundational step in noise reduction. Typically, the Pan–Tompkins methodology encompasses five distinct stages, primarily focused on the detection and enhancement of QRS complexes [18]. However, for our specific classification purposes, we opt to utilize solely the bandpass filtering stage of the algorithm. This decision streamlines our preprocessing pipeline, bypassing the later QRS-specific stages that are not necessary for our requirements, thereby optimizing computational resources.

Originally, the filters in PTA are calibrated for a 5–12-Hz bandpass range, assuming a standard ECG sampling rate of 200 Hz. However, the ECG signals in the MIT-BIH dataset are sampled at 360 Hz, prompting a design decision: either recalibrate the filters or downsample the data. We chose to preserve the original sampling rate to retain high-frequency components, which potentially enhances frequency resolution. Consequently, we adjusted the bandpass range to 9–21.6 Hz to maintain equivalence with the original PTA behavior.

Our testing confirmed the effectiveness of this approach. Despite the shift in frequency range, the classification efficacy remained uncompromised compared to filters explicitly designed for the 360-Hz sampling rate. This finding supports our decision to use the original PTA bandpass filter design with only minimal modifications, leveraging its low computational footprint. By capitalizing on simple operations, such as addition and bit shifting, the preprocessing overhead remains minimal and well-suited for hardware-constrained environments.

The filtered QRS samples are used as classification features for COIN and are further encoded using the RRT encoding technique in the final stage of preprocessing. It is important to note that no preprocessing steps were synthesized into the chip; the logic synthesis and FlexIC implementation were based solely on the COIN inference model. This deliberate omission ensures that the evaluation reflects the raw computational efficacy and integration of COIN within FlexIC constraints.

C. Cross-Validation and Model Selection

The MIT-BIH dataset presents several challenges for model validation. First, it is extremely imbalanced and limited in size, comprising data from only 48 patients. Furthermore, the distribution of arrhythmia types across these patients is uneven, with certain categories concentrated in only a few individuals. This issue is clearly demonstrated in Fig. 5, which shows the distribution of the number of beats per patient for each class. The plots reveal the variability and concentration of beats per

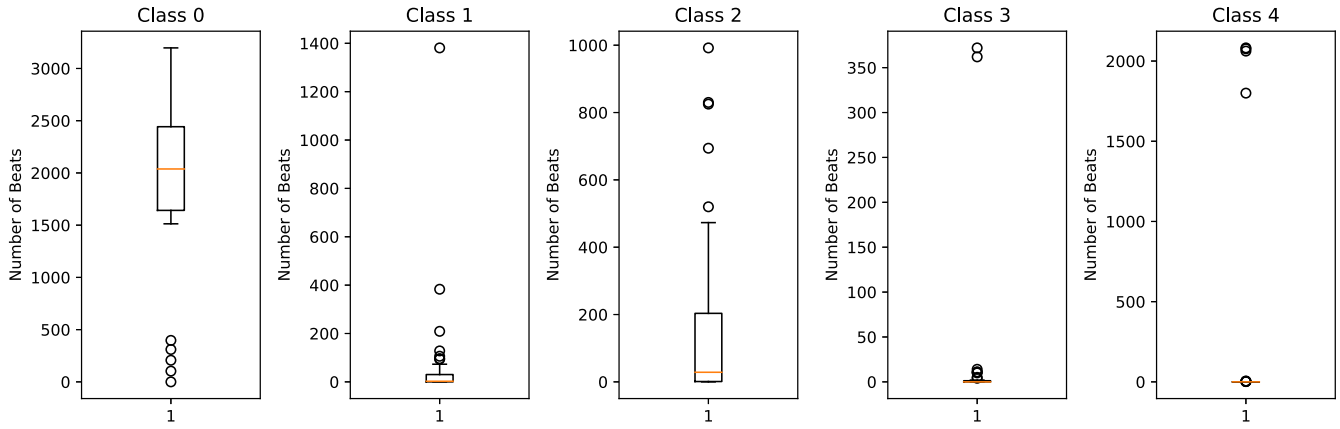

Fig. 5. Box plots illustrating the distribution of the number of beats for each arrhythmia class (Classes 0–4) across 48 patients in the MIT-BIH dataset. The plots reveal the variability and concentration of beats per class among the patients, highlighting the dataset's imbalance and the challenge of class distribution. Outliers and the spread of the data underscore the uneven distribution of arrhythmia types, indicating the necessity of a careful approach to data partitioning and validation methodology.

class among the patients, highlighting the necessity of a careful approach to data partitioning and the validation methodology.

Given these factors, a standard k-fold cross-validation would not yield robust results, as it would be nearly impossible to partition the data without including samples from the same patient in both training and test sets simultaneously. To address this issue, we adopted a Monte Carlo cross-validation strategy. In each round, we created new splits of approximately 70% of the data for training, 15% for validation, and 15% for testing. Despite our classifier being binary, maintaining class stratification was crucial for robust performance. These proportions were targeted but not always achievable due to patientwise separation. Therefore, we allowed flexibility in validation rounds; when creating a fold, data from the same class could occupy up to 50% of the total test or validation set, except for the normal beat class (the most abundant), which was limited to 15%.

We conducted 50 Monte Carlo cross-validation rounds, each with new patient-based partitions, to mitigate statistical bias. During the grid search phase, we simultaneously evaluated various hyperparameter configurations, including address size (ranging from 8 to 16), thermometer resolution (from 2 to 8), and the presence or absence of data augmentation. For each configuration, we explored ten distinct random mappings and selected the configuration with the most favorable outcome. This exploration led us to conclude that an address size of 8 and a thermometer resolution of 2 yielded favorable results.

The final model, featuring 741 minterms, was selected based on its alignment with average performance indicators to achieve a balance between computational efficiency and diagnostic accuracy. This model's exceptional specificity of 0.9983 demonstrates its capability to precisely flag nonarrhythmic beats, while its sensitivity of 0.6862 indicates potential for further enhancement of arrhythmic detection.

Addressing this sensitivity-specificity tradeoff is vital for arrhythmia detection due to the implications of false negatives. Unlike probabilistic classifiers, COIN does not generate output probabilities but rather produces discrete scores for each discriminator and the class with the highest score is selected as the final decision. Although weighted normalization could be applied to explore different operating points, this work adopts the standard COIN decision rule for consistency and simplicity.

The results of the Monte Carlo cross-validation, partially shown in Fig. 6, reveal a complex landscape between model size (number of minterms) and AUC scores (performance accuracy). Notably, the data do not exhibit a clear linear correlation between complexity and effectiveness, underscoring the nuanced challenge of optimizing model design. This highlights the importance of a strategic search for a model that harmonizes minimal model size with satisfactory diagnostic performance—particularly critical when implementing these models in FlexIC technology, where efficiency and compactness are paramount.

D. Synthesis

The synthesis of the selected model using FlexIC technology resulted in a prelayout synthesized design comprising a total of 2549 NAND2-equivalent gates, with no RAM or LUT macros and only small sequential state. This model was obtained using the hyperparameter configuration discussed in Section IV-C. To investigate the relationship between model complexity and hardware requirements within FlexIC technology, we conducted a synthesis analysis for a range of models generated using different mapping functions, all derived from the same dataset split as the selected model.

Based on the synthesis area reports, sequential logic accounts for approximately 36.7% of the total area, while the combinational portion represents 63.3%.

This synthesis analysis, whose results are shown in Fig. 7, unveils a distinct and linear relationship between the number of minterms in a model and the gate count required for its FlexIC technology implementation. The linear trend highlighted in these results demonstrates a consistent pattern: models characterized by fewer minterms tend to necessitate a smaller number of gates for their physical realization. This observation

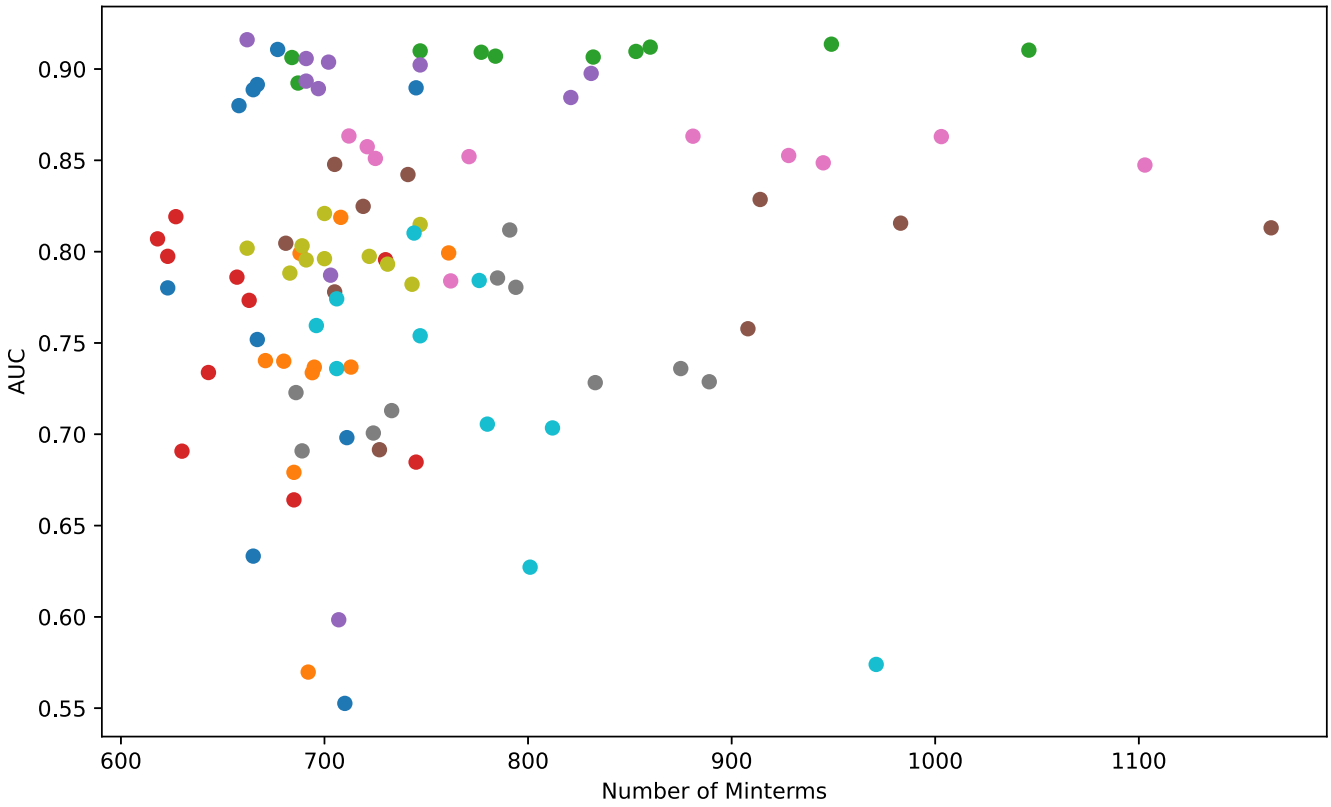

Fig. 6. AUC scores plotted against the number of minterms (related to model size) for models in ten separate runs. The absence of a discernible pattern underscores the necessity of searching for a compact yet effective model suitable for FlexIC technology deployment.

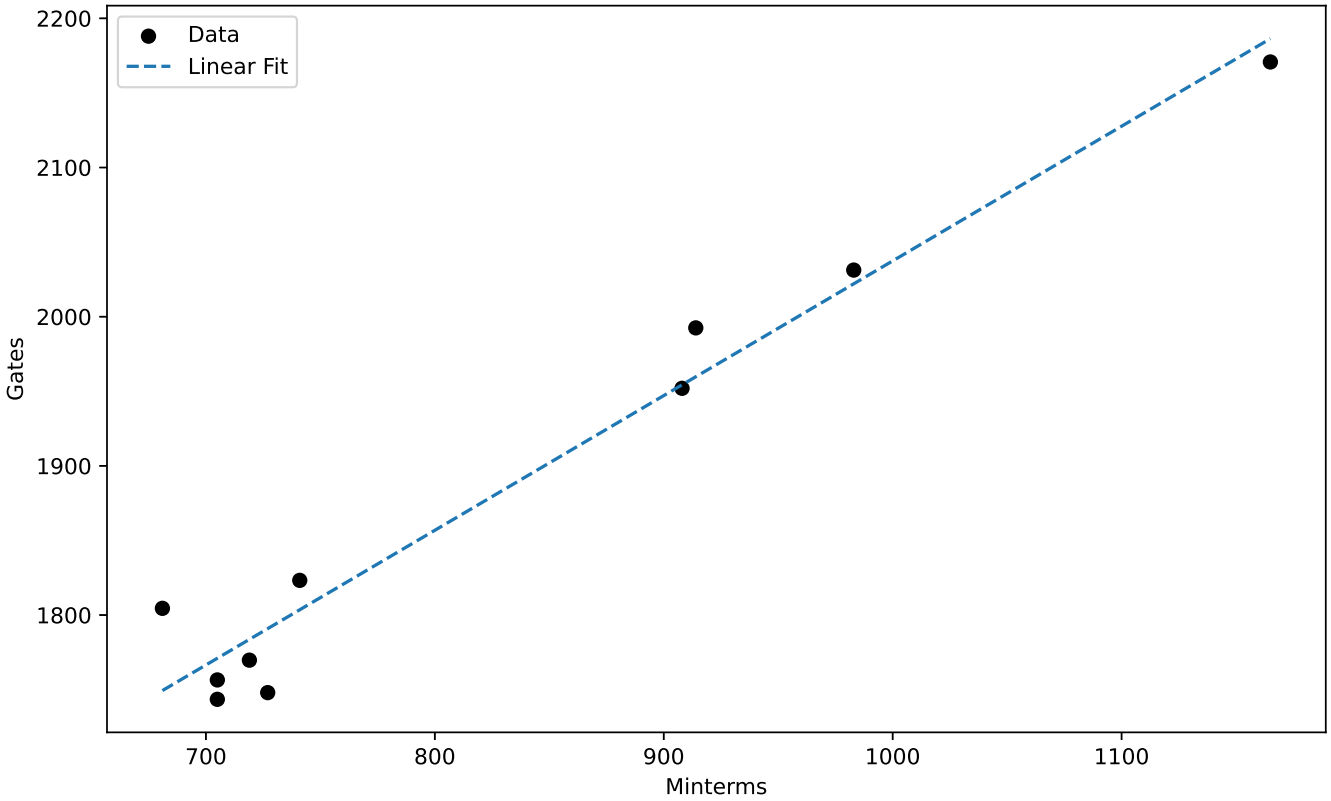

Fig. 7. This plot illustrates the linear relationship between the number of minterms and the corresponding gate count required for models synthesized using FlexIC technology. Each point represents a model derived from a mapping function, showcasing how models with fewer minterms lead to a reduction in gate count, emphasizing the feasibility of gate-aware model optimization.

is critical as it paves the way for a gate-aware training regimen, enabling the deliberate development of models that naturally align with the specific constraints and objectives of the project and technology at hand.

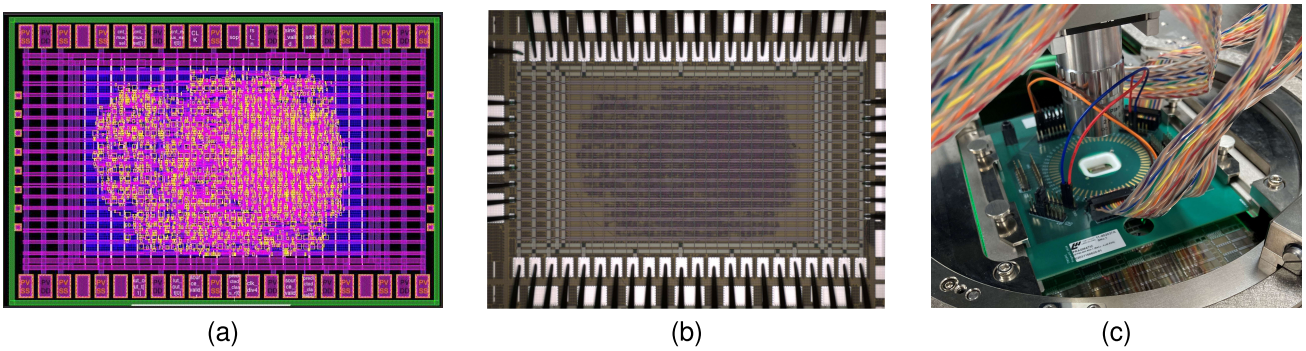

Fig. 8. Arrhythmia detector chip development. (a) GDS layout, (b) SEM micrograph of the chip, showing the probe card contact points for testing, and (c) probe card testing a wafer on the wafer probe station.

V. Proof-of-Concept Implementation of the Arrhythmia Detector With FlexIC

We detail the transition from netlist to a fully functional GDSII file, followed by the fabrication of our arrhythmia detector using FlexIC technology. This process illustrates the feasibility and reliability of our automated design methodology. The arrhythmia detecting COIN-based FlexIC was implemented and fabricated by Pragmatic's 0.6-μm process.

After place-and-route, the design operates at a clock frequency of 100 kHz and integrates 5706 NAND2-equivalent gates, along with 15 functional I/O pins in addition to power pins. The gate count increased compared to the synthesis results presented in Section IV-D, primarily due to routing congestion and additional buffering required to manage the long combinational paths inherent to the COIN architecture. Following synthesis, comprehensive physical design steps were undertaken, with back-annotation performed after each phase to ensure accurate gate-level simulation results. The final design occupied a core area of 6×4 mm^2 and exhibited a power consumption of 7.26 mW at 2.7 V. The GDS layout of the implemented FlexIC is shown in Fig. 8(a).

The chip's fabrication and testing phases are illustrated in Fig. 8. We tested the fabricated chip using a probe card for direct die contact, as depicted in Fig. 8(b) and (c), which limited our test setup to 6.25 kHz due to high capacitive loading. This constraint is specific to the test harness and does not reflect the chip's designed operating frequency of 100 kHz, which was met in gate-level simulations with positive timing slack.

The preprocessing and mapping stages are performed off-chip, while the COIN inference logic is fully implemented on the flexible device. During operation, the 376-bit input vector can be loaded serially into the chip over 752 clock cycles, after which the decision is generated within 15 additional clock cycles. At an operating frequency of 100 kHz, this would correspond to a decision rate of approximately 130.4 inferences per second.

This testing, involving 306 inferences, yielded predictions that perfectly matched the simulations, therefore preserving the same accuracy as the model showed before. Fig. 9 showcases these results, highlighting a correct synchronization between input and output signals during the test sequence. This includes the final stages of the input data stream (sink_valid and addr) and the corresponding outputs (source_valid and predicted_class). Notably, the slow rise of the predicted_class signal can be attributed to the capacitive loading from the logic analyzer and the limited drive strength of the output buffers.

VI. Discussion

The proof-of-concept journey described in Sections IV and V, from model training to chip testing, demonstrates the agility, effectiveness, and adaptability of our design methodology for FlexIC-based ML systems, paving the way for future advancements in wearable health monitoring technologies.

Our methodology supports handling more complex designs than previous FlexIC implementations and integrates model size optimization early in the training phase, ensuring efficient performance within FlexIC constraints. The design cycle is remarkably swift, with data-to-netlist completion in a single day and synthesis in a few days, as pins and architecture remain unchanged. This efficiency makes the entire chip design process possible in under a week.

We prioritized first-pass success by opting for sparse gate placement and redundant control circuits, minimizing the risk of layout errors and ensuring reliability, even though this led to suboptimal area utilization. Future designs can enhance area efficiency by refining these initial choices. Additionally, albeit our approach is not a one-size-fits-all solution, it enables leveraging the advantages of plastic technology to produce competitive designs along desired metrics, such as accuracy.

Comparing the proposed design with previous implementations for arrhythmia classification using the MIT-BIH dataset is not straightforward. First, to the best of our knowledge, no prior work has implemented this task on FlexIC technology, making direct comparisons in terms of power, area, and operating frequency infeasible. Moreover, our evaluation adopts a patientwise validation approach (leaving one patient out), which better reflects real deployment conditions in medical monitoring but typically leads to lower reported accuracies compared to conventional cross-validation methods that mix patient data between training and test sets. Therefore, accuracy comparisons with prior studies should be interpreted with caution. Additionally, most implementation articles do not disclose gate counts or NAND2-equivalent metrics, which would enable a more technology-independent comparison of design compactness. Among the few that do, the work by Chen and Juan [19] reports a gate count of 23.6 K—considerably larger

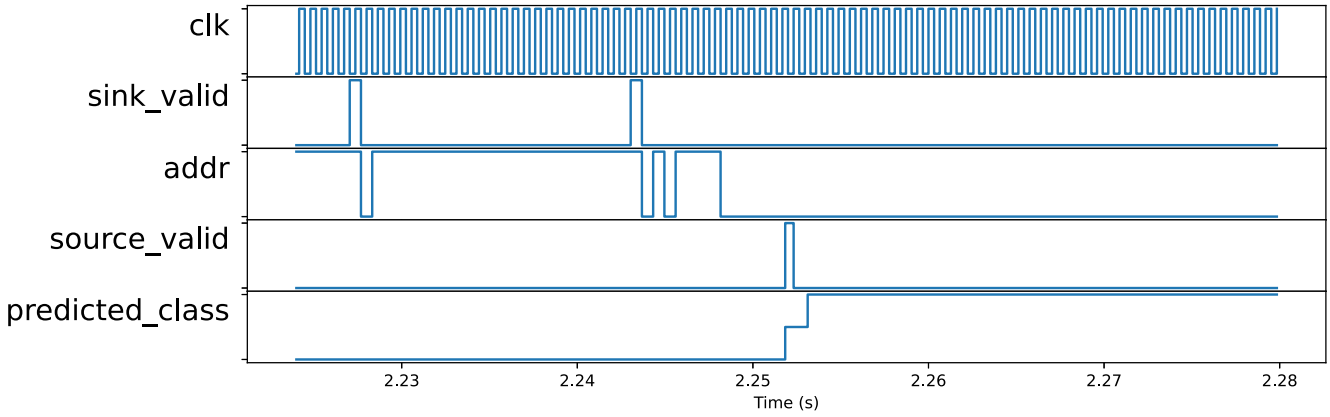

Fig. 9. Waveform of the tested chip, displaying the synchronization of input and output signals during a test sequence. The figure shows the final stages of the input data stream (sink_valid and addr) and the inference outputs (source_valid and predicted_class).

than the 5.7 K gates required by our COIN-based FlexIC, underscoring the compactness of the proposed approach.

VII. Conclusion

In this study, we introduced an advanced ML framework for wearable technology, leveraging the synergistic capabilities of COIN and FlexICs. This fusion enhances the computational efficiency and adaptability of wearable devices while adhering to sustainable technology development principles. Demonstrated through a proof-of-concept model for arrhythmia detection, our approach underscores the technology's potential in critical health monitoring, highlighting its rapid design process, efficiency, and real-world applicability.

The novel integration of COIN with FlexICs provides a flexible, cost-effective, and scalable solution suitable for implementing medium-complexity tasks required for wearable monitoring. Future work will focus on extending automation to the physical layout phase, aiming to reduce or eliminate manual intervention currently required in the place-and-route and GDSII generation steps.

Code Availability

The code used in this study is available at https://github.com/lasdi/flexwnn

Acknowledgment

Any opinions, findings, conclusions, or recommendations are those of the authors and not of the funding agencies.

References

[1] (2024). *Flexible ICs*. [Online]. Available: https://www.pragmaticsemi.com/flexible-ics

[2] E. Ozer et al., "Bespoke machine learning processor development framework on flexible substrates," in *Proc. IEEE Int. Conf. Flexible Printable Sensors Syst. (FLEPS)*, Jul. 2019, pp. 1–3.

[3] E. Ozer et al., "A hardwired machine learning processing engine fabricated with submicron metal-oxide thin-film transistors on a flexible substrate," *Nature Electron.*, vol. 3, no. 7, pp. 419–425, Jul. 2020.

[4] E. Ozer et al., "Binary neural network as a flexible integrated circuit for odour classification," in *Proc. IEEE Int. Conf. Flexible Printable Sensors Syst. (FLEPS)*, Aug. 2020, pp. 1–4.

[5] E. Ozer et al., "Malodour classification with low-cost flexible electronics," *Nature Commun.*, vol. 14, no. 1, p. 777, Feb. 2023.

[6] E. Ozer et al., "A custom-designed atrial fibrillation detection hardware on a flexible substrate," in *Proc. IEEE Int. Conf. Flexible Printable Sensors Syst. (FLEPS)*, Jun. 2024, pp. 1–4.

[7] K. Iordanou et al., "Low-cost and efficient prediction hardware for tabular data using tiny classifier circuits," *Nature Electron.*, vol. 7, no. 5, pp. 405–413, Apr. 2024, doi: 10.1038/s41928-024-01157-5.

[8] I. D. S. Miranda et al., "COIN: Combinational intelligent networks," in *Proc. IEEE 34th Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Jul. 2023, pp. 27–28.

[9] I. D. S. Miranda et al., "LogicWiSARD: Memoryless synthesis of weightless neural networks," in *Proc. IEEE 33rd Int. Conf. Appl.-Specific Syst., Archit. Processors (ASAP)*, Jul. 2022, pp. 19–26.

[10] I. Aleksander, M. De Gregorio, F. França, P. Lima, and H. Morton, "A brief introduction to weightless neural systems," in *Proc. 17th Eur. Symp. Artif. Neural Netw. (ESANN)*, Apr. 2009, pp. 299–305.

[11] I. Aleksander, W. V. Thomas, and P. A. Bowden, "WISARD·a radical step forward in image recognition," *Sensor Rev.*, vol. 4, no. 3, pp. 120–124, Mar. 1984. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/eb007637/full/html

[12] A. T. L. Bacellar et al., "Distributive thermometer: A new unary encoding for weightless neural networks," in *Proc. 30th Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, Mar. 2022, pp. 1–10.

[13] Z. Susskind et al., "Weightless neural networks for efficient edge inference," in *Proc. 31st Int. Conf. Parallel Archit. Compilation Techn.*, Oct. 2022, pp. 279–290.

[14] Y. Umuroglu et al., "FINN: A framework for fast, scalable binarized neural network inference," in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, Feb. 2017, pp. 65–74, doi: 10.1145/3020078.3021744.

[15] (2024). *Pragmatic Semiconductor*. [Online]. Available: https://www.pragmaticsemi.com/

[16] A. Ahamed, P. Huang, J. Young, A. Gallego-Schmid, R. Price, and M. P. Shaver, "Technical and environmental assessment of end-of-life scenarios for plastic packaging with electronic tags," *Resour., Conservation Recycling*, vol. 201, 2024, Art. no. 107341.

[17] 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, Mar. 2001.

[18] 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.

[19] Y. Chen and Y.-C. Juan, "Very-large-scale integration implementation of a convolutional neural network accelerator for abnormal heartbeat detection," *Electron. Lett.*, vol. 56, no. 7, pp. 330–331, 2020.