Invited Paper: Feature-to-Classifier Co-Design for Mixed-Signal Smart Flexible Wearables for Healthcare at the Extreme Edge

Maha Shatta*, Konstantinos Balaskas§, Paula Carolina Lozano Duarte*, Georgios Panagopoulos‡‡,
Mehdi B. Tahoori*, Georgios Zervakis§

*Karlsruhe Institute of Technology, DE, §University of Patras, GR, ‡‡National Technical University of Athens, GR

*{maha.shatta, paula.duarte, mehdi.tahoori}@kit.edu, §{kompalas, zervakis}@ceid.upatras.gr, ‡‡gepanago@mail.ntua.gr

Abstract—**Flexible Electronics (FE) offer a promising alternative to rigid silicon-based hardware for wearable healthcare devices, enabling lightweight, conformable, and low-cost systems. However, their limited integration density and large feature sizes impose strict area and power constraints, making ML-based healthcare systems–integrating analog frontend, feature extraction and classifier–particularly challenging. Existing FE solutions often neglect potential system-wide solutions and focus on the classifier, overlooking the substantial hardware cost of feature extraction and Analog-to-Digital Converters (ADCs)–both major contributors to area and power consumption. In this work, we present a holistic mixed-signal feature-to-classifier co-design framework for flexible smart wearable systems. To the best of our knowledge, we design the first analog feature extractors in FE, significantly reducing feature extraction cost. We further propose an hardware-aware NAS-inspired feature selection strategy within ML training, enabling efficient, application-specific designs. Our evaluation on healthcare benchmarks shows our approach delivers highly accurate, ultra-area-efficient flexible systems–ideal for disposable, low-power wearable monitoring.**

Index Terms—**Co-design, Flexible Electronics, Machine Learning, Feature Extraction, Healthcare Wearables**

I. INTRODUCTION

In recent years, the demand for advanced healthcare applications has grown significantly, driven by the increasing need for continuous and personalized monitoring of an individual's physiological state [1]. As healthcare systems shift from reactive to proactive models, the ability to continuously monitor patients outside clinical settings has become increasingly important, with a growing interest towards wearable devices [2]–[4]. These devices capture physiological signals (e.g., electrodermal activity (EDA)) during everyday activities, enabling real-time, healthcare tracking at the extreme edge.

Significant efforts have focused on the algorithmic development of wearable devices, particularly with the use of Machine Learning (ML) algorithms [5]–[8]. Targeting tasks such as stress, heart or respiration monitoring [9]–[11], works span a wide range of ML models which analyze diverse biosignals for real-time health monitoring [5]–[7], [12]–[15]. However, little attention has been paid to the underlying hardware, or to potential co-design opportunities, since the sensing/analog frontend is typically designed in isolation from the ML classifier. Most commercial wearables still rely on rigid, silicon-based microcontrollers, which hinder comfort and skin conformability; their general-purpose design wastes

energy, reducing battery life, while high manufacturing costs limit affordability and adoption [16].

Flexible Electronics (FE) have emerged as a compelling alternative to rigid silicon-based platforms for wearable healthcare. Built on lightweight, conformable substrates, Flexible Integrated Circuits (FlexICs) conform to body contours, improving comfort and long-term wearability [17], [18]. FE also enable ultra–low-cost, disposable systems for clinical and consumer use. Their fabrication sidesteps many silicon constraints (cleanrooms, packaging), enabling fast, low-cost–even portable–production [19]–[21], and offers environmental gains (lower water/energy use and carbon footprint) over silicon processes [21]. Together, these features position FE as a sustainable, scalable, and user-friendly foundation for next-generation accessible wearable health monitoring systems.

The integration of flexible components into wearables mainly targets mechanically bendable bio-sensors, capable of conformably acquiring diverse signals in real time (e.g., skin temperature) [17], [22]–[24]. On the algorithmic side, FlexICs have been used for implementing ML classification [25], [26], even targeting applications such as malodour detection [27]. However, FE are constrained by large feature sizes and limited integration density–stemming from their low-cost fabrication processes [28], [29]–which lead to increased area and power requirements, making it challenging to implement complex flexible ML-based systems. Furthermore, integrating FE circuits with commercial off-the-shelf silicon components can be cumbersome–e.g., for the analog front end–necessitating co-design and co-fabrication of the entire system, from sensor and analog interface to classifier, on the same substrate.

Within these constraints, a typical FE-based healthcare monitoring system integrates flexible sensors for physiological signal acquisition, quantization via Analog-to-Digital Converters (ADCs), and on-device feature extraction and ML classification. Typically, such systems are mostly implemented in digital logic (Fig. 1a). Although each component plays a critical role, most existing FE approaches concentrate solely on the design and optimization of either the sensor interface or the ML classifier [22]–[24], [30]–[39]. Acknowledging this limitation, [40]–[43] co-design the analog interface with the classifier, achieving notable savings in both the interface and the overall system. However, the feature extractor, which plays a vital role in increasing classification performance and main-
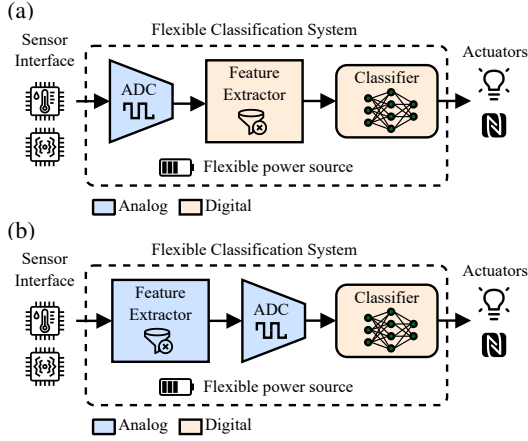
Fig. 1: Abstract overview of a flexible classification system, with feature extractors in the (a) digital, or (b) analog domain

taining acceptable hardware requirements, is yet overlooked by the state of the art. In fact, as we demonstrate in [26], feature extractors form the area bottleneck in such flexible systems (Fig. 1a) and can incur prohibitive area and power costs, potentially challenging the feasibility of the entire system.

In our work, we address these challenges with an automated feature-to-classifier co-design framework for ultra-area-efficient mixed-signal flexible healthcare systems. Our approach jointly optimizes all core components—including ADCs, feature extractors, and classifier—aiming to reduce the hardware overheads of ADC and feature extraction. First, to mitigate its elevated area overhead, we implement feature extraction directly in the analog domain–as shown in Fig. 1b– and, to the best of our knowledge, design the first analog feature extractors in FlexIC technology. Next, we design a Successive Approximation Register (SAR) ADC optimized for our system, where all analog features are quantized and stored in buffers for processing by the classifier. For classification, we employ shallow digital Multi-Layer Perceptrons (MLPs)—tailored to the application, i.e., bespoke designs—which can deliver top accuracy within realistic hardware overheads [26]. Finally, we propose a hardware-aware feature selection technique embedded within MLP training, which minimizes feature extraction cost while maximizing accuracy.

Our novel contributions within this work are as follows:

1) We design, for the first time, analog feature extractors in FlexIC targeting conformable classification systems.
2) We propose the first holistic feature-to-classifier co-design and co-optimization framework for mixed-signal flexible systems, where feature extraction is implemented in the analog and classification in the digital domain, to balance cost and accuracy, along with a novel hardware-aware feature selection embedded within MLP training.
3) We evaluate our work on relevant healthcare datasets and demonstrate that our co-design framework enables flexible systems with high accuracy and ultra-low area and energy-per-inference requirements, making them ideal for low-cost, conformal, and disposable healthcare wearables.
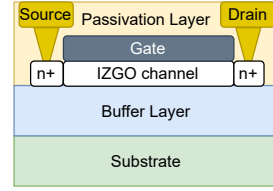

Fig. 2: Process schematic illustrating photolithographic patterning on IGZO semiconductors and metallic interconnects.

II. BACKGROUND ON FLEXICS

Pragmatic's FlexIC technology enables the fabrication of mechanically flexible circuits on ultra-thin polyimide substrates using Indium Gallium Zinc Oxide (IGZO) Thin-Film Transistors (TFTs) [21], [28], [44]. IGZO TFTs are manufactured using low-temperature photolithography and cost-effective equipment, eliminating the need for rigid silicon wafers, high-temperature processes, and protective packaging, as illustrated in Fig 2. This streamlined process significantly reduces environmental impact, lowers production costs, and shortens fabrication time—from over 30 weeks in traditional silicon technology to just a few days for FlexICs [21]. The produced circuits are mechanically robust, bendable to a radius as small as $3\,\mathrm{mm}$, and well-suited for conformal, disposable electronics in healthcare applications.

Despite these advantages, IGZO-based FlexICs face key limitations compared to CMOS technology. Their relatively large feature sizes (600–800nm) and low integration density lead to increased power consumption with tight area requirements [21], [28], [45]. Due to the absence of stable p-type transistors, FE are restricted to n-type devices and unipolar logic. As a result, resistive-load NMOS logic incurs high–especially static–power consumption and high circuit latencies, leading to elevated energy demands as well. As a result, current FlexIC designs are limited to moderate complexity—typically only a few thousand gates—and require careful hardware-software co-design to simplify logic, reduce gate count, and limit memory elements, which remain costly in FE [28], [33]. Such constraints highlight the need for specialized, lightweight design methodologies—especially in the healthcare domain where area, energy autonomy, and flexibility are critical.

III. MOTIVATION

Feature selection is often treated purely from an algorithmic perspective, without consideration for its hardware implications [8], leading to designs that, in FE, may be infeasible for the underlying area and power constraints. Indicatively, we present an example in Fig. 3 using the WESAD stress-monitoring dataset [4], where we perform feature selection with the statistical Fisher score algorithm, similar to [8]. We then train a MLP classifier on the extracted features and design the digital circuits that implement the required feature extractors and the resulting MLP. As shown in Fig. 3, the feature extraction consumes the central portion of the total area, at 46%, which rises to 74% when accounting for ADC costs. In addition, ADCs account for 51% of the total
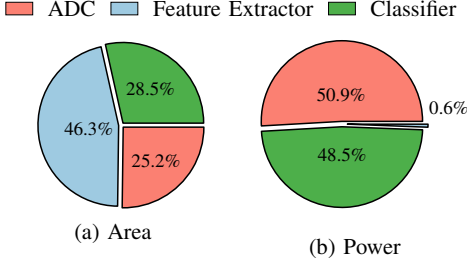
(a) Area (b) Power

Fig. 3: Area and power breakdown of an MLP-based flexible system on the WESAD dataset [4]. We observe that the digital feature extractors consume a major portion of the total area, whereas the ADCs constitute the system's power bottleneck.

power consumption of the system. *We therefore conclude that focusing solely on the ML classifier is suboptimal and can challenge the design feasibility of such flexible systems. To achieve the utmost hardware efficiency–as required in FE to even enable feasibility–a holistic feature-to-classifier optimization is mandatory.*

IV. FLEXIBLE MIXED-SIGNAL SYSTEM ARCHITECTURE

An abstract overview of our proposed system architecture is illustrated in Fig. 4. Assuming an array of biosensors, the analog feature extraction circuits process the incoming sensor signals and compute the required statistics (features) over a predefined window size. Each feature for each sensor is computed on dedicated hardware, with all features computed concurrently. At the end of the timing window, the analog features (outputs of the analog feature extraction circuits) are multiplexed through the SAR ADC, and their quantized values are stored in a buffer for subsequent processing by the digital MLP classifier. A small digital control logic block (e.g., a few counters, a decoder) orchestrates the entire operation.

A. Analog Feature Extractors

In [26], we conducted a comprehensive exploratory study on the implications and hardware overheads of feature extraction in the design of ML-based flexible healthcare wearable systems. Fig. 5 shows the frequency of the most commonly selected features in the accuracy–area Pareto-optimal designs identified in [26]. As observed, simple features such as maximum, minimum, mean, and sum appear more frequently in Pareto-optimal solutions and are always present in the most accurate ones. Therefore, in our work, we design the analog equivalents of these four statistical functions in FlexIC technology, and integrate them into our co-design process.

1) Maximum (Max): The Max block uses a peak-detector to capture and hold the highest value of the input signal. As shown in Fig. 6(a), it comprises a diode and a hold capacitor. When $v_{\text{in}} > v_C + V_{TH}$, the diode conducts and charges the capacitor to the new peak; when $v_{\text{in}} < v_C$, the diode is reverse-biased, blocking current and preserving the stored peak.

Since the FlexIC PDK provides only n-type transistors and no discrete diodes, we implement the diode function
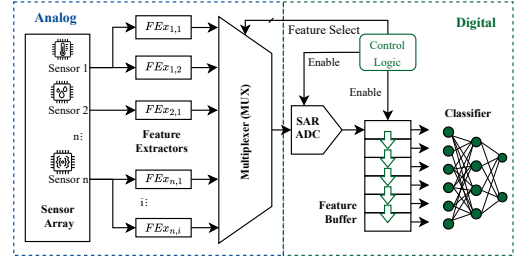

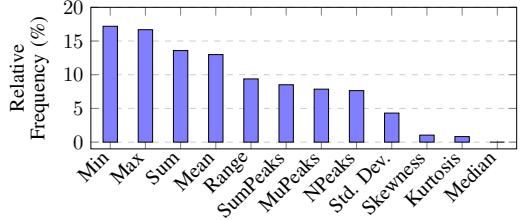Fig. 4: Overview of our proposed flexible mixed-signal system architecture

Fig. 5: Most frequent features selected in the accuracy–area Pareto-optimal flexible classifiers from [26] across various healthcare datasets

with a diode-connected n-type transistor (gate tied to drain). This introduces a headroom requirement: conduction requires $V_{GS} \geq V_{TH}$, so any path through this element incurs an effective forward drop of roughly V_{TH}, which reduces available signal swing and sets a minimum rectifiable amplitude. In addition, *leakage*, dominated by subthreshold conduction when $V_{GS} < V_{TH}$ causes residual discharge that degrades low-level accuracy; the droop rate is approximately $\dot{v}_C \approx -I_{\text{leak}}/C$. We mitigate these effects by using longer-channel devices.

Capacitor sizing trades hold time against tracking speed: larger C increases the hold time (lower $|\dot{v}_C|$) but slows response to rapid input changes, whereas smaller C enables faster tracking at the cost of increased droop and reduced hold time. A windowing switch is included to reset the function. It isolates the hold capacitor at each window boundary and clears the previous state before the next window. For the Max configuration, the reset discharges the hold capacitor to the low reference V_l, after which it can charge within the window to track the maximum.

2) Minimum (Min): The Min block is a valley detector that captures and holds the lowest value of the input over each observation period. As in Fig. 6(a), the diode–capacitor network is oriented to pull the capacitor node down when the input decreases. When $v_{\text{in}} < v_C - V_{TH}$, the diode-connected n-type transistor conducts and discharges the capacitor to the new minimum; when $v_{\text{in}} > v_C$, the device is reverse-biased, blocking current and preserving the stored minimum.

Similar to the Max implementation, the diode is implemented with a diode-connected n-type transistor. This introduces a headroom condition: updates occur only when the input drop exceeds V_{TH}, limiting sensitivity to small dips. In
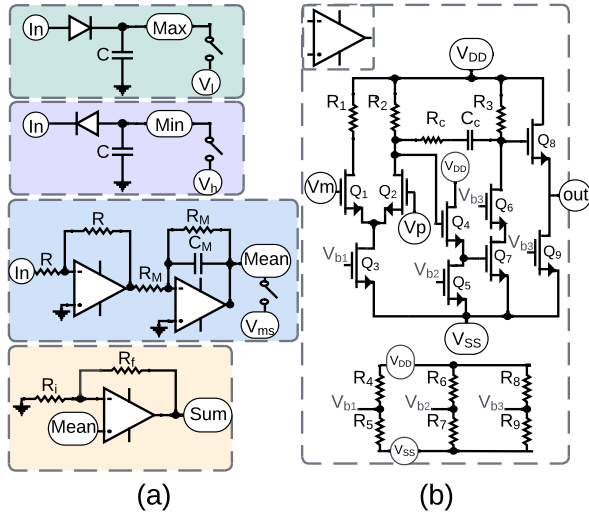
Fig. 6: (a) Analog feature circuits of Max, Min, Mean, and Sum. (b) Op-amp implementation in the FlexIC PDK

addition, leakage causes the stored value to drift from the true minimum over time, at a rate approximately $|\dot{v}_C| \approx I_{\text{leak}}/C$.

A windowing switch is included, as in the Max design. In the Min configuration, however, the reset pre-charges the hold capacitor to the upper reference V_h, after which it can only discharge within the window to track the minimum.

3) Mean: The mean calculation circuit (Fig. 6(a)) is based on an op-amp integrator, which continuously accumulates the input signal, summing values over a defined period according to the integration equation. This accumulated output approximates the mean when normalized by the integration period T or scaled by $\frac{1}{R_M C_M}$, analogous to dividing a discrete sum by the number of samples. An inverter adjusts the polarity of the input signal, which is then inverted again by the integrator, ensuring the final output matches the expected mean value.

$$\text{Mean } (\mu) = \frac{1}{N} \sum_{i=1}^{N} x_i \approx \frac{1}{R_M C_M} \int_0^T x(t)\, dt, \quad (1)$$

where x_i represents discrete input samples, N is the sample count, $x(t)$ is the continuous-time input signal, T is the integration period, and R_M and C_M are the resistor and capacitor values in the integrator circuit, respectively.

The windowing switch also performs a reset at each window boundary. For the Mean block, we reset to V_{ms}, the mid-scale (virtual-ground) level of the op-amp output range (typically $V_{ms} \approx (V_{\text{DD}} + V_{\text{SS}})/2$), so each window starts centered.

4) Sum: The sum operation is similar to the mean calculation but scaled by the number of samples N. Leveraging this, we scale the mean output to approximate the sum. However, since analog circuits are limited by the op-amp's output swing, the scaling factor is chosen to keep the resulting output within the op-amp's acceptable operating range.

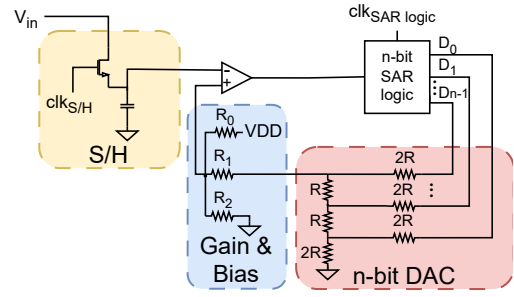$$\text{Sum} = \mu \times N, \quad N = 1 + \frac{R_f}{R_i}. \quad (2)$$

Fig. 7: Schematic of an n-bit SAR ADC

The scaling is implemented using a non-inverting amplifier configuration, as shown in Fig 6(a), where the gain follows the relationship in (2). By selecting appropriate resistor values R_f and R_i, the circuit scales the mean output to produce the sum. Since the Sum is derived directly from the Mean output, it does not require a separate reset for each window.

5) op-amp: Designing the op-amp in FlexIC presents challenges, as the technology includes only n-type transistors and lacks p-type devices. Despite this limitation, a two-stage op-amp with an additional buffer stage is implemented, as shown in Fig. 6(b). The op-amp consists of a differential amplifier stage, which includes a pair of n-type transistors forming the differential pair (Q_1, Q_2), along with a tail-bias transistor Q_3 and passive resistive loads R_1 and R_2. A common-source amplifier follows this stage to enhance the gain of the op-amp. In this configuration, two transistors are used: one to stabilize the operating point and set the current of the common-source amplifier Q_6, and the other to serve as the input transistor Q_7, receiving the signal from the preceding stage. To ensure proper operation of Q_7, we add a level-shifting stage using transistors Q_4 and Q_5 to set the node at the required DC bias level. The output of the differential amplifier is coupled to the input of the common-source stage through a coupling capacitor C_c and resistor R_c, which help maintain stability and improve phase margin. To ensure low output impedance, a buffer stage is included with two transistors Q_8 and Q_9. It isolates the op-amp's high-gain stages from the load, reducing the risk of loading effects and improving driving capability.

B. Flexible SAR ADC

Various ADC designs have been explored in FE, including Flash [46], Sigma-Delta [47], Binary Search [40], and SAR [30]. SAR ADC is attractive due to its moderate hardware complexity, scalability to different resolutions, and low static power dissipation. Therefore, we design a custom SAR ADC fully tailored to our application requirements (e.g., speed, precision, input range), thereby maximizing hardware efficiency in our systems.

A SAR ADC determines the n-bit digital output code in n cycles. In each cycle, the SAR logic sets the current bit and drives a digital-to-analog converter (DAC). The DAC output is compared to the sampled analog input by a comparator, and the SAR logic updates the code accordingly. If the comparator
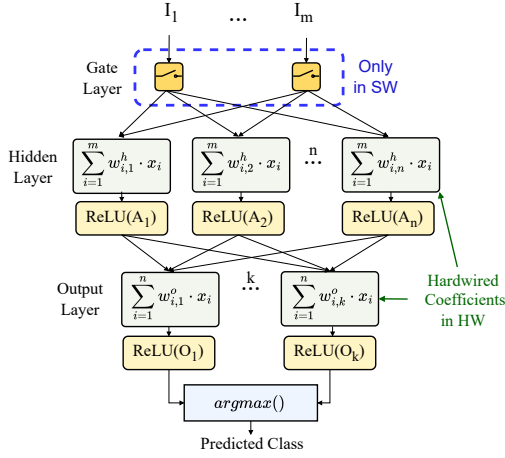
Fig. 8: Bespoke fully-parallel MLP design

Fig. 9: Algorithmic overview of our proposed hardware-aware co-design framework

output is high, the current bit remains set; otherwise, it is cleared before moving to the next bit. Once the LSB has been processed, the resulting SAR logic value is the ADC output.

An n-bit schematic of our designed SAR ADC is shown in Fig. 7. The control logic is implemented in Verilog RTL as two n-bit registers and mapped to PragmatIC's FlexIC standard digital cell library. This cell library is based on a resistive-load logic architecture, which uses a fixed resistor for the pull-up. The DAC is implemented as an R–$2R$ ladder network, producing 2^n discrete voltage levels. A subsequent gain-and-bias stage scales and offsets the DAC output to align with the output range of the analog feature circuits. Although our ADC can operate significantly faster, we observe that across all healthcare datasets examined in Section VI, a conversion time of 0.5ms is sufficient for real-time monitoring.

C. Flexible Digital MLP Classifier

The digital part of our system involves an MLP classifier (see Fig. 1b), due to its effectiveness in delivering high accuracy at relevant applications [26], [35], [36]. Leveraging the ultra-low manufacturing and non-recurring engineering (NRE) costs of flexible electronics (FE), we implement our digital MLP as a fully-parallel bespoke architecture [25], [33], [36], as shown in Fig. 8. Bespoke ML circuits hardwire the trained coefficients into the circuit, enabling significant area savings compared to conventional designs [48], and facilitating further logic simplification through constant propagation during synthesis. Fully-parallel MLPs instantiate one hardware neuron per software neuron, with each containing one multiplier per trained weight, followed by a precision-optimized adder tree for product accumulation. All neurons operate concurrently, eliminating the need for costly memory elements in FE. Such architectures are favorable to unstructured pruning, with direct hardware savings due to the lack of folding. We later demonstrate how unstructured pruning-aware retraining is embedded into our framework for further area reductions.

V. FEATURE-TO-CLASSIFIER CO-DESIGN FRAMEWORK

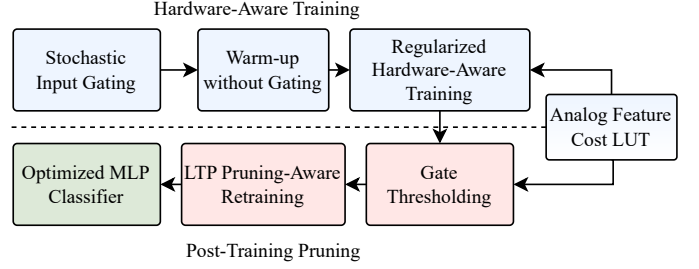In this section, we present our automated co-design framework for training the flexible MLP classifier. An algorithmic overview of our framework is presented in Fig. 9. It first introduces a novel hardware-aware feature selection technique embedded within training (Section V-A), aiming to reduce the analog feature extractor costs and retain high classification accuracy within a unified differentiable step. Then, leveraging our system's architecture, pruning with retraining steps follow (Section V-B) to further maximize area efficiency.

A. Differentiable Feature Selection & Training

Our feature selection mechanism is inspired by Neural Architecture Search (NAS) methods [49], which typically focus on layer or block selection. Adapted for our purposes, our approach operates at the feature level, embedding a differentiable stochastic gating layer into the input stage of an MLP, as shown in Fig. 8. By integrating cost-aware regularization into the gating mechanism, the technique enables end-to-end optimization of both accuracy and analog feature extraction cost within the training process.

1) Stochastic Feature Gating: Assuming all input features for all sensors $\mathbf{x} \in \mathbb{R}^d$, we introduce a trainable gating vector $\mathbf{z} \in [0, 1]^d$ applied as a multiplicative factor to the input:

$$\tilde{\mathbf{x}} = \mathbf{z} \odot \mathbf{x} \tag{3}$$

Each gate z_i represents the inclusion probability of feature x_i and is modeled as a stochastic binary variable. Features with lower gate values (i.e., close to zero) contribute negligibly during training and can therefore be considered non-important. To enable gradient-based optimization, this binary behavior is approximated using the Concrete (Gumbel–Sigmoid) distribution [50]. During training, gates are stochastically sampled as:

$$z_i = \text{clip}(s_i, 0, 1), \tag{4}$$

$$s_i = \frac{1}{1 + \exp\left(-\frac{\log u_i - \log(1-u_i) + \log \alpha_i}{\gamma}\right)}, \tag{5}$$

where $\log \alpha_i$ is the trainable parameter (i.e., logit) controlling the openness of the gate, $u_i \sim \mathcal{U}(0,1)$ is a uniform random variable, and γ is a hyperparameter controlling the relaxation. As γ approaches 0, the distribution becomes more discrete, yielding gate values closer to $0, 1$ while still maintaining differentiability for optimization. By progressively lowering γ, we can gradually separate essential features from insignificant ones based on their corresponding gate values.

After training has converged, the gates provide interpretable importance scores per feature. Specifically, the stochastic sampling above is replaced by simple deterministic gating:

$$z_i = \sigma(\log \alpha_i), \tag{6}$$

where $\sigma(\cdot)$ is the logistic function mapping logits to probabilities in $(0, 1)$. Thus, during inference the gating vector $\mathbf{z}$ is fixed and modulates the contribution of each feature.

The gating layer is placed directly between the input and the first hidden layer (see Fig. 8), ensuring that all subsequent computations operate only on the modulated input $\tilde{\mathbf{x}}$, thereby minimally impacting the network's architecture.

2) Area-Informed Regularized Training: In order to introduce hardware-awareness into the training and embed the feature extraction cost into the stochastic gates, we employ regularization by adding a cost-aware term to our loss function. Specifically, we collect the area of our analog feature extraction circuits (see Section IV-A) in a Look-Up Table (LUT), which is then proportionally added to the training loss.

Let $\mathbf{c} \in \mathbb{R}^d$ be the vector of per-feature area costs. Accounting for the gated contribution of each feature, the expected cost $\mathcal{L}_{\text{cost}}$ is combined with the task loss as follows:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \cdot \mathcal{L}_{\text{cost}}, \quad \mathcal{L}_{\text{cost}} = \sum_{i=1}^{d} \sigma(\log \alpha_i) \cdot c_i, \tag{7}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss, and λ a scalar hyperparameter controlling the accuracy-cost trade-off. By directly embedding feature extraction costs into the optimization objective, the model is incentivized to discover feature subsets that are jointly optimal for accuracy and hardware efficiency.

Since regularization is known for destabilizing training [51], we employ a warm-up phase of k epochs during which gradients through the gate layer are detached, keeping them fixed at their initialized values:

$$z_i = \text{stop\_gradient}(z_i) \quad \text{if epoch} < k. \tag{8}$$

This allows the MLP weights to adapt to the objective loss ($\mathcal{L}_{\text{CE}}$) before stochastic gating–and therefore input destabilization–is introduced.

3) Feature Pruning via Gate Removal: After convergence, we perform feature selection by removing gates whose value is below a threshold τ:

$$\hat{z}_i = \begin{cases} 1 & \text{if } z_i > \tau \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

This thresholding produces a sparse and hardware-efficient input representation, preserving high-importance features while eliminating those with low contribution to accuracy. By progressively iterating over various thresholds $\tau \in [0, 1]$, we obtain a Pareto-front of networks that trade off accuracy and feature extraction cost in a fully automated, hardware-aware manner.

B. Lottery-Ticket Pruning-Aware Retraining

Following feature selection and training, where the analog frontend costs are optimized, we aim to reduce the significant contribution of the digital classifier to the total system area. Due to its bespoke, fully parallel architecture, unstructured pruning can be directly exploited in this context to reduce the number of parameters, thereby yielding high hardware gains by eliminating multipliers and minimizing the area of accumulators [36]. To that end, we adopt pruning-aware retraining in the form of Lottery-Ticket Pruning (LTP), inspired by the Lottery Ticket Hypothesis [52], which stipulates that within a dense, randomly-initialized neural network, there exists a sparse subnetwork capable of matching the original performance when trained in isolation.

Let $\mathbf{W} \in \mathbb{R}^n$ denote the vectorized parameters of the trained MLP, and let $\mathbf{m} \in \{0, 1\}^n$ be a binary pruning mask, where $m_j = 0$ indicates removal of weight W_j. We perform iterative magnitude pruning by updating $\mathbf{m}$ according to:

$$m_j = \begin{cases} 0 & \text{if } |W_j| \leq \kappa_t \\ 1 & \text{otherwise,} \end{cases} \tag{10}$$

where κ_t is the pruning threshold at iteration t, chosen to achieve a target sparsity $s_t \in [0, 1]$. After each pruning step, the remaining weights are reset to their pre-training values $\mathbf{W}_0 \odot \mathbf{m}$, and the sparse subnetwork is retrained for a reduced number of epochs, progressively increasing s_t until the desired final sparsity is reached.

VI. RESULTS & ANALYSIS

A. Experimental Setup

We evaluate our proposed framework on 3 popular healthcare benchmarks related to the use-case of stress-monitoring, where physiological data are extracted from biosensors to infer the stress levels of individuals. Specifically, we use the WESAD dataset [4]–the most popular and common dataset for stress-monitoring applications–alongside Stress-In-Nurses [53], and Stress Predict dataset (or SPD) [54]. Though, our approach can be seamlessly extended to any relevant far-edge and/or healthcare application and dataset.

Input data are in floating-point format and normalized within the $[0, 1]$ range. Feature extraction is simulated at high level in Python using a non-overlapping sliding window of $1\,\text{s}$. K-fold cross-validation is used on all considered datasets, using 80% of subjects for training, and the rest 20% for test on unseen individuals' data. The accuracy is reported on the test set hereafter. For training, we use Tensorflow and consider an MLP with one hidden layer of 100 neurons, such that high enough computational accuracy can be achieved, while any redundancy from the large neuron count can be removed by pruning-aware retraining. The Adam optimizer with a learning rate of 0.001 is used for 50 epochs of training (10 for the pruning-aware retraining), along with early-stopping. The regularization parameter λ (see (7)) and γ (see (5)) undergo hyperparameter tuning via Bayesian optimization within empirical ranges. Finally, the gate pruning thresholds (τ in

TABLE I: Dimensions of op-amp components

Component	Size
Q_1 to Q_7	W = 20 μm, L = 1.2 μm
Q_8	W = 200 μm, L = 0.6 μm
Q_9	W = 1 μm, L = 1.2 μm
R_1, R_2 = 368.5 kΩ	W = 2.1 μm, L = 4.4 μm
R_3 = 837.6 kΩ	W = 2.1 μm, L = 10 μm
R_4 = 2.3 MΩ	W = 1 μm, L = 12.3 μm
R_5 = 1 MΩ	W = 1 μm, L = 5.3 μm
R_6 = 15.4 MΩ	W = 0.6 μm, L = 48.2 μm
R_7 = 10 MΩ	W = 0.6 μm, L = 28.5 μm
R_8, R_9 = 20 MΩ	W = 0.6 μm, L = 61.4 μm
R_C = 50 kΩ	W = 50 μm, L = 15 μm
C_C = 0.95 pF	W = 14 μm, L = 15 μm

TABLE II: Design properties of the SAR ADC

Component	Element	Size
S/H	Transistor	w = 15μm, l = 600 nm
	Capacitor	cap = 2 pF
DAC	Resistor	R = 1.5 MΩ
	Resistor	R_0, R_2 = 6 MΩ
	Resistor	R_1 = 4.5 MΩ
Comparator	We use the op-amp presented in Table I	
SAR Digital Logic	Std-cell based design with Gen-3 FlexIC	

(9)) are swept within $\{0.01, 0.05, 0.1, 0.2, 0.5\}$. MLP weights are quantized post-training to 8 bits, while input features are quantized to 4 bits, matching the precision that we use for the ADC. Such low input/ADC precision is common in FE applications [26], [27], [36] and is sufficient to maintain high accuracy while incurring low analog interfacing costs.

For analog and mixed-signal simulations, we use the Cadence Spectre simulator with the Gen3 PragmatIC FlexIC 1.0.0 PDK [44]. The supply voltage is set to 3 V. For digital synthesis, timing, and power simulations (e.g., for the MLP), Synopsys Design Compiler, VCS, and PrimeTime are used. Synthesized designs are mapped to PragmatIC's Gen-3 FlexIC PDK characterized at 3 V [44]. Our classifiers are synthesized at a relaxed clock period using the `compile_ultra` command and target area optimization. A base clock of 10 kHz is considered, and our systems target real-time performance, i.e., producing a stress prediction every 1 s. Specifically, after the last sample of the window is received, our system completes the classification (feature-to-MLP) within less than 20 ms. Power-gating is used for idle components.

The area of the analog circuits is estimated from the pre-layout device dimensions (W–L) of the components. The op-amp sizing is listed in Table I, Q_8 is intentionally wider in the buffer stage to increase the output-swing range. The opamp's power consumption is measured in a closed-loop configuration. The ADC is designed for 4-bit quantization, with transistor and resistor dimensions provided in Table II. Although R_0, R_1, and R_2 should be theoretically equal to keep the signal within 1–2V, the inherent circuit attenuation requires adjusting these values. To compensate, we set R_0 and R_2 larger than R_1. By this adjustment, the DAC output ranges from 0.98V to 1.95V.

TABLE III: Op-amp performance metrics

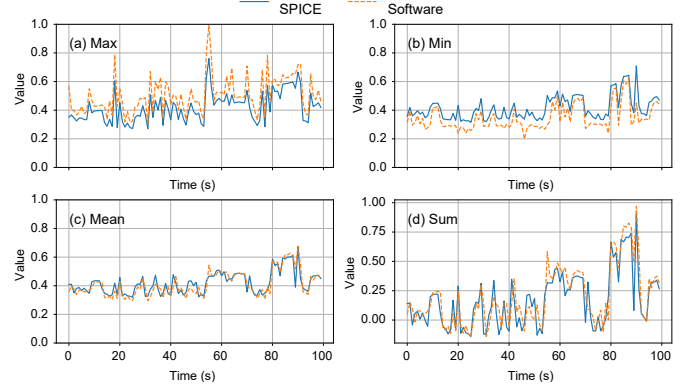| Metric	Value
Gain (after buffer stage)	28.6 dB
Unity Gain Bandwidth	902 kHz
Phase Margin	58°
Slew Rate (Rising / Falling)	0.9 V/μs, 1.4 V/μs
Input Offset Voltage	−30 mV
Input Voltage Range	−1.5 V, 1.2 V
Output Voltage Range	−1.5 V, 0.8 V
Output Impedance	230 Ω
Dual Power Supply	±1.5 V
Power Consumption	71 μW
Area	0.0014 mm^2

Fig. 10: Output of the Max/Min/Mean/Sum analog circuits (blue) versus ideal feature value from software (orange) for the SPD dataset and the accelerometer x-axis signal

TABLE IV: Evaluation of analog feature extraction circuits

Feature	NMSE	Power (μW)	Area (mm^2)
Max	0.005	0.44	0.0020
Min	0.004	0.44	0.0020
Mean	0.003	155	0.0084
Sum	0.002	232	0.0099

B. Analog Components Evaluation

The op-amp characterization results are summarized in Table III, showing a good enough gain of 28.6dB. The power consumption of our SAR ADC is measured to be 81.4μW and its area is as low as 0.02mm^2.

Next, we evaluate our analog features as standalone statistical circuits. Fig. 10 presents a comparison between the SPICE analog output and the software reference value, for one sensor (accelerometer x-axis) from the SPD dataset. For *Sum*, to avoid early saturation, the circuit used a partial scaling by a representative factor N; the residual factor was applied in software so the final trace reflects the intended sum for a fair SPICE–software comparison. It is also adjusted to fall within a similar range to the other three features. As shown in Fig. 10, the analog output closely follows the ideal software output. Table IV reports the accuracy and hardware measurements of our analog circuits as function approximators. For the former, we consider the Normalized Mean Squared Error (NMSE) between our circuit outputs–obtained through SPICE simulations–and ideal software-computed features, averaged

across the test set of all considered benchmarks (WESAD, SPD, and Stress-In-Nurses). As shown, all features achieve low NMSE, indicating small absolute deviations. In terms of hardware metrics, *Max* and *Min* share identical hardware overheads since they employ the same topology, differing only in diode orientation, whereas *Mean* and *Sum* incorporate an op-amp–based stage with its feedback/bias network, resulting in relatively higher power and area, as expected. Indicatively, the reported area overheads are 97% lower compared to the respective digital feature extraction circuits.

C. System Evaluation

In this section, we evaluate our complete classification systems—comprising all components: analog feature extractors, ADC, and MLP classifier—obtained through our feature-to-classifier co-design. We focus on three key metrics: accuracy, which defines the system's performance; area, which is critical in FE applications due to the limited number of integrated devices [28]; and energy per inference, which determines battery lifetime. For reference, we compare our solutions against [26], where a statistical-based feature selection method is used, combined with a brute-force exploration over the number of selected features, pruning sparsity, and weight quantization precision to identify area-efficient solutions. Hereafter, we refer to SoA 4F as the solution from [26] when considering only the *min*, *max*, *mean*, and *sum* features (as in our work) in their feature selection process, and to SoA AF as the respective solutions when all twelve statistical features (see Fig.5) are considered in [26]. In [26], both feature extractors and MLPs are implemented in digital. Table V presents the comparative results for all approaches and datasets.

As shown in Table V, our circuits deliver high classification accuracy with well-contained hardware costs. Specifically, power consumption reaches only 20.3 mW–well below the capabilities of existing printed batteries (e.g., a 30 mW Molex printed battery)–while energy per inference remains below 1 μJ. The area ranges from 0.06 mm^2 to 27.43 mm^2, *while achieving* 73% *to* 89% *classification accuracy and real-time monitoring within an accessible, mechanically flexible, and conformable healthcare wearable.* Moreover, it is noteworthy that, despite any inaccuracies introduced by our analog feature extraction circuits, the classification accuracy remains within 3% of the purely software-based floating-point results. The highest accuracy degradation is observed in SPD, most likely because it requires 21 features, whereas the other datasets require fewer than 5. Finally, as shown in Table V, unlike our motivation study, our co-design, combined with our area-efficient analog features, effectively reduces the feature extraction area to a negligible portion of the overall cost.

Compared to [26], our solutions achieve significantly higher accuracy (around 8% on average), despite the accuracy drop due to analog feature extraction, due to our integration of feature selection directly into MLP training, whereas [26], as typically done [8], employs a pre-training statistical feature selection to reduce the complexity of design space exploration. Note that the designs of [26] are purely digital,

TABLE V: Comparative in-depth analysis of our proposed approach against the state of the art [26]

Technique	Software Accuracy (%)	Circuit Accuracy (%)	Total Area (mm^2)	Feature Area (mm^2)	Total Power (mW)	Energy/ Inference (μJ)
WESAD						
SoA AF[1][26]	68.35	68.35	216.5	7.243	166.0	9.31
SoA 4F[2][26]	73.95	73.95	241.3	3.581	178.7	10.5
Ours	83.66	81.52	4.092	0.004	2.963	0.056
Stress-In-Nurses						
SoA AF[1][26]	72.69	72.69	17.3	7.99	7.161	0.313
SoA 4F[2][26]	74.46	74.46	66.43	3.579	30.62	2.5
Ours	89.12	89.08	0.198	0.004	0.136	0.008
SPD						
SoA AF[1][26]	67.60	67.60	105.5	7.971	73.03	3.83
SoA 4F[2][26]	67.32	67.32	86.79	1.937	62.03	3.28
Ours	73.42	70.41	27.57	0.078	20.3	0.763

[1]Using all 12 features from [26]. [2]Using only *min*, *max*, *mean*, and *sum*.

achieving the expected software accuracy. Still, our solutions achieve higher accuracy than both SoA AF—demonstrating that limiting our solutions to the respective four selected analog features did not compromise accuracy—and SoA 4F, showing that incorporating our regularizer during training to minimize feature extraction cost likewise does not affect the achievable accuracy. Moreover, our analog implementation of feature extraction allows for orders of magnitude lower area compared to digital features–of more than $600\times$–partly due to our effective feature selection scheme and due the area-efficiency of analog features. Overall, our solutions achieve an average area reduction of $48\times$ compared to the most efficient solution of [26], while energy per inference is reduced by $70\times$, enabling significantly longer energy autonomy compared to purely digital implementations.

VII. Conclusion

FlexICs offer a compelling alternative to rigid silicon for wearable healthcare devices, enabling lightweight, conformable, and low-cost systems. However, their large feature sizes and limited integration density impose strict area and power constraints that challenge full ML implementations. In this work, we introduced the first feature-to-classifier co-design framework for mixed-signal flexible systems, combining custom analog feature extractors with an optimized SAR ADC and a hardware-aware in-training feature selection. Across multiple healthcare benchmarks, our approach achieves real-time operation with high accuracy, energy consumption below 1uJ per inference, and practical area requirements.

REFERENCES

[1] M. M. Khan and M. Alkhathami, "Anomaly detection in iot-based healthcare: machine learning for enhanced security," *Scientific Reports*, vol. 14, no. 1, p. 5872, 2024.

[2] K. Mahato *et al.*, "Hybrid multimodal wearable sensors for comprehensive health monitoring," *Nature Electronics*, vol. 7, no. 9, 2024.

[3] S. Jeong *et al.*, "Exploiting boosting in hyperdimensional computing for enhanced reliability in healthcare," in *Design, Automation & Test in Europe Conference (DATE)*, 2025, pp. 1–7.

[4] P. Schmidt, A. Reiss, R. Dürichen, C. Marberger, and K. V. Laerhoven, "Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection," in *International Conference on Multimodal Interaction*, 2018, pp. 400–408.

[5] A. Arsalan, M. Majid, S. M. Anwar, and U. Bagci, "Classification of Perceived Human Stress using Physiological Signals," in *Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBC)*, 2019.

[6] A. Kumar, K. Sharma, and A. Sharma, "Hierarchical deep neural network for mental stress state detection using IoT based biomarkers," *Pattern Recognit. Lett.*, vol. 145, pp. 81–87, 2021.

[7] S. A. H. Aqajari *et al.*, "GSR Analysis for Stress: Development and Validation of an Open Source Tool for Noisy Naturalistic GSR Data," *ArXiv*, vol. abs/2005.01834, 2020.

[8] S. Jiang, F. Firouzi, K. Chakrabarty, and E. B. Elbogen, "A resilient and hierarchical iot-based solution for stress monitoring in everyday settings," *IEEE Internet of Things Journal*, vol. 9, no. 12, 2022.

[9] N. E. Haouij, J.-M. Poggi, S. Sevestre-Ghalila, R. Ghozi, and M. Jaïdane, "Affectiveroad system and database to assess driver's attention," in *ACM Symposium on Applied Computing*, 2018, pp. 800–803.

[10] A. Logacjov, K. Bach, A. Kongsvold, H. B. Bårdstu, and P. J. Mork, "Harth: a human activity recognition dataset for machine learning," *Sensors*, vol. 21, no. 23, p. 7853, 2021.

[11] D. Bhattacharya *et al.*, "Coswara: A respiratory sounds and symptoms dataset for remote screening of sars-cov-2 infection," *Scientific data*, vol. 10, no. 1, p. 397, 2023.

[12] A. Tazarv *et al.*, "Personalized stress monitoring using wearable sensors in everyday settings," in *Int. Conf IEEE Engineering in Medicine & Biology Society (EMBC)*, 2021, pp. 7332–7335.

[13] G. Boateng and D. Kotz, "Stressaware: An app for real-time stress monitoring on the amulet wearable platform," in *IEEE MIT Undergraduate Research Technology Conference (URTC)*, 2016, pp. 1–4.

[14] A. Golgouneh and B. Tarvirdizadeh, "Fabrication of a portable device for stress monitoring using wearable sensors and soft computing algorithms," *Neural Computing and Applications*, vol. 32, no. 11, 2020.

[15] N. Attaran, A. Puranik, J. Brooks, and T. Mohsenin, "Embedded low-power processor for personalized stress detection," *IEEE Trans. Circuits Syst. II*, vol. 65, no. 12, pp. 2032–2036, 2018.

[16] V. Mishra *et al.*, "Continuous detection of physiological stress with commodity hardware," *ACM Trans. Comput. Healthcare.*, vol. 1, no. 2, pp. 1–30, 2020.

[17] W. Gao *et al.*, "Fully integrated wearable sensor arrays for multiplexed in situ perspiration analysis," *Nature*, vol. 529, pp. 509–514, 01 2016.

[18] W. Heng, S. Solomon, and W. Gao, "Flexible electronics and devices as human–machine interfaces for medical robotics," *Advanced Materials*, vol. 34, no. 16, p. 2107902, 2022.

[19] N. Bleier *et al.*, "Printed microprocessors," in *Annu. Int. Symp. Computer Architecture (ISCA)*, jun 2020, pp. 213–226.

[20] N. Bleier *et al.*, "Flexicores: low footprint, high yield, field reprogrammable flexible microprocessors," in *International Symposium on Computer Architecture (ISCA)*, 2022, p. 831–846.

[21] E. Ozer *et al.*, "Bendable non-silicon risc-v microprocessor," *Nature*, pp. 1–6, 2024.

[22] X. Wang, Z. Liu, and T. Zhang, "Flexible Sensing Electronics for Wearable/Attachable Health Monitoring," *Small*, vol. 13, no. 25, 2017.

[23] Y. Yang and W. Gao, "Wearable and flexible electronics for continuous molecular monitoring," *Chem. Soc. Rev.*, vol. 48, pp. 1465–1491, 2019.

[24] S. Yoon, J. K. Sim, and Y.-H. Cho, "A flexible and wearable human stress monitoring patch," *Scientific reports*, vol. 6, no. 1, p. 23468, 2016.

[25] E. Özer *et al.*, "A hardwired machine learning processing engine fabricated with submicron metal-oxide thin-film transistors on a flexible substrate," *Nature Electronics*, vol. 3, pp. 1–7, 07 2020.

[26] F. Afentaki *et al.*, "Exploration of low-power flexible stress monitoring classifiers for conformal wearables," in *International Symposium on Low Power Electronics and Design (ISLPED)*, 2025.

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

[28] M. B. Tahoori, E. Ozer, G. Zervakis, K. Balaskas, and P. Pal, "Computing with printed and flexible electronics," *IEEE European Test Symposium (ETS)*, 2025.

[29] J. Henkel *et al.*, "Approximate computing and the efficient machine learning expedition," in *Int. Conf. on Computer-Aided Design (ICCAD)*, 2022, pp. 1–9.

[30] F. Alkhalil *et al.*, "Flexible sar adc with resistive dac for conformable on-body sensing applications," in *2022 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2022, pp. 110–114.

[31] A. Anzanpour *et al.*, "Self-awareness in remote health monitoring systems using wearable electronics," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 2017, pp. 1056–1061.

[32] C.-M. Chen *et al.*, "Towards wearable and flexible sensors and circuits integration for stress monitoring," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 8, pp. 2208–2215, 2020.

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

[34] K. Iordanou *et al.*, "Low-cost and efficient prediction hardware for tabular data using tiny classifier circuits," *Nature Electronics*, 2024.

[35] G. Armeniakos, G. Zervakis, D. Soudris, M. B. Tahoori, and J. Henkel, "Co-design of approximate multilayer perceptron for ultra-resource constrained printed circuits," *IEEE Trans. Comp.*, pp. 1–8, 2023.

[36] A. Kokkinis, G. Zervakis, K. Siozios, M. B. Tahoori, and J. Henkel, "Enabling printed multilayer perceptrons realization via area-aware neural minimization," *IEEE Transactions on Computers*, 2024.

[37] F. Afentaki *et al.*, "Bespoke approximation of multiplication-accumulation and activation targeting printed multilayer perceptrons," in *Int. Conf. Computer Aided Design (ICCAD)*, 2023, pp. 1–9.

[38] K. Balaskas, G. Zervakis, K. Siozios, M. B. Tahoori, and J. Henkel, "Approximate decision trees for machine learning classification on tiny printed circuits," in *Int. Symp. Quality Electronic Design*, 2022, pp. 1–6.

[39] A. Kokkinis *et al.*, "Hardware-aware automated neural minimization for printed multilayer perceptrons," in *Design Automation and Test in Europe (DATE)*, 2023.

[40] P. C. Lozano Duarte, F. Afentaki, G. Zervakis, and M. Tahoori, "Design and in-training optimization of binary search adc for flexible classifiers," in *Asia and South Pacific Design Automation Conference*, 2025.

[41] F. Afentaki, P. C. L. Duarte, G. Zervakis, and M. B. Tahoori, "Reducing adc front-end costs during training of on-sensor printed multilayer perceptrons," *IEEE Embedded Syst. Lett.*, vol. 16, no. 4, 2024.

[42] G. Armeniakos *et al.*, "On-sensor printed machine learning classification via bespoke adc and decision tree co-design," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2024, pp. 1–6.

[43] V. Mrazek *et al.*, "Evolutionary approximation of ternary neurons for on-sensor printed neural networks," in *International Conference on Computer-Aided Design (ICCAD)*, 2025.

[44] Pragmatic, "Flexic Platform Gen3," https://www.pragmaticsemi.com/foundry/flexic-platform-gen-3, 2025.

[45] J. Biggs *et al.*, "A natively flexible 32-bit arm microprocessor," *Nature*, vol. 595, pp. 532–536, 2021.

[46] A. Jamshidi-Roudbari, P.-C. Kuo, and M. K. Hatalis, "A flash analog to digital converter on stainless steel foil substrate," *Solid-State Electronics*, vol. 54, no. 4, pp. 410–416, 2010.

[47] C. Garripoli and et al., "15.3 an a-igzo asynchronous delta-sigma modulator on foil achieving up to 43db snr and 40db sndr in 300hz bandwidth," in *IEEE Int. Solid-State Circuits Conference (ISSCC)*, 2017.

[48] G. Armeniakos, G. Zervakis, D. Soudris, M. B. Tahoori, and J. Henkel, "Cross-layer approximation for printed machine learning circuits," in *Design Automation and Test in Europe (DATE)*, 2022, pp. 190–195.

[49] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *International Conference on Learning Representations (ICLR)*, 2018.

[50] X. Geng *et al.*, "How does selective mechanism improve self-attention networks?" in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2986–2995.

[51] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li, "Learning structured sparsity in deep neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[52] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

[53] S. Hosseini *et al.*, "A multimodal sensor dataset for continuous stress detection of nurses in a hospital," *Scientific Data*, vol. 9, no. 1, 2022.

[54] T. Iqbal *et al.*, "Stress monitoring using wearable sensors: A pilot study and stress-predict dataset," *Sensors*, vol. 22, no. 21, p. 8135, 2022.