This article is accepted for publication in *IEEE Design, Automation & Test in Europe in Europe (DATE), 2026*

Design and Optimization of Mixed-Kernel Mixed-Signal SVMs for Flexible Electronics

Florentia Afentaki*°, Maha Shatta†°, Konstantinos Balaskas*, Georgios Panagopoulos‡‡, Georgios Zervakis*, Mehdi B. Tahoori†
*University of Patras, GR †Karlsruhe Institute of Technology, DE ‡‡National Technical University of Athens, GR
*{afentaki, kompalas, zervakis}@ceid.upatras.gr †{maha.shatta, mehdi.tahoori}@kit.edu ‡‡gepanago@mail.ntua.gr

***Abstract*—Flexible Electronics (FE) have emerged as a promising alternative to silicon-based technologies, offering on-demand low-cost fabrication, conformality, and sustainability. However, their large feature sizes severely limit integration density, imposing strict area and power constraints, thus prohibiting the realization of Machine Learning (ML) circuits, which can significantly enhance the capabilities of relevant near-sensor applications. Support Vector Machines (SVMs) offer high accuracy in such applications at relatively low computational complexity, satisfying FE technologies' constraints. Existing SVM designs rely solely on linear or Radial Basis Function (RBF) kernels, forcing a trade-off between hardware costs and accuracy. Linear kernels, implemented digitally, minimize overhead but sacrifice performance, while the more accurate RBF kernels are prohibitively large in digital, and their analog realization contains inherent functional approximation. In this work, we propose the first mixed-kernel and mixed-signal SVM design in FE, which unifies the advantages of both implementations and balances the cost/accuracy trade-off. To that end, we introduce a co-optimization approach that trains our mixed-kernel SVMs and maps binary SVM classifiers to the appropriate kernel (linear/RBF) and domain (digital/analog), aiming to maximize accuracy whilst reducing the number of costly RBF classifiers. Our designs deliver 7.7% higher accuracy than state-of-the-art single-kernel linear SVMs, and reduce area and power by $108\times$ and $17\times$ on average compared to digital RBF implementations.**

***Index Terms*—Flexible Electronics, Mixed-Signal**

I. Introduction

Flexible Electronics (FE) have emerged as a promising alternative to silicon-based computing, offering attractive properties such as mechanical flexibility, non-toxicity, and conformality, whilst enabling low-cost fabrication and sustainability. These characteristics make FE suitable for low-cost far-edge applications, such as wearable healthcare monitoring [1]–[6], and smart packaging [7], [8]. However, FE is fundamentally constrained by large feature sizes, limited device counts, and low integration density, which result in increased power and area consumption, limiting scalability and efficiency. Thus, implementing complex circuits in FE, such as machine learning (ML) classifiers, remains challenging. Yet, ML-based classification is essential for sensor-driven applications that translate raw signals into meaningful decisions, which are central to FE use cases [9].

To address these limitations, most state-of-the-art FE designs rely on approximate computing—exploiting the inherent error tolerance of ML, trading accuracy for efficiency [10]–[16]. Typically, multi-layer perceptrons (MLPs) are employed, which require aggressive approximation to mitigate their large area/power overhead, and thus introduce large accuracy loss.

Support vector machines (SVMs) are a favorable alternative to MLPs for the simpler classification tasks targeted by FE, offering modest model complexity, reduced area/power requirements, and competitive accuracy across a range of relevant applications [17], [18]. At their core, SVMs typically rely on either linear or radial basis function (RBF) kernels. RBF kernels provide higher classification accuracy [19], [20]—in our experiments, 7.7% higher accuracy on average than linear kernels—but incur prohibitively large area and power overheads for realization in the digital domain. Thus, RBF-based SVMs in FE can only include cost-effective analog circuits, which, however, contain inherent functional approximation. On the other hand, linear kernels offer reduced computational complexity and hardware cost, but primarily result in lower achievable accuracy. Thus, restricting flexible SVMs to a single kernel type inevitably forces a compromise between classification accuracy and hardware efficiency.

In this work, we present the first mixed-kernel, mixed-signal SVM tailored for FE. Specifically, we combine linear and RBF kernels in a unified design, aiming to achieve high accuracy while instantiating only a small number of RBF units. Leveraging that multiclass SVMs can be decomposed into binary classifiers, we implement the RBF classifiers as analog FlexICs—directly computing on sensory inputs—whereas linear ones as digital designs. Operating in the subthreshold regime, analog RBF blocks are based on functionally approximate Gaussian kernels, enabling ultra-low-power and high area efficiency. Thus, our mixed-kernel design improves accuracy by selectively deploying RBF beyond linear-only baselines. It also shifts the otherwise prohibitive digital RBF into the analog domain, reducing both power and area. Furthermore, we propose a separation-based exploration strategy that enables the automatic identification of the optimal kernel type per binary classifier—by assessing its accuracy contribution—and facilitates mixed-kernel SVM training optimized for high accuracy while minimizing the number of costly RBF classifiers. Our designs achieve 7.7% higher accuracy on average compared to the state-of-the-art single-kernel linear SVMs, with $108\times$ and $17\times$ average lower area and power, respectively, compared to all-RBF digital solutions.

°Authors contributed equally to this work.

The main contributions of this work are as follows:

1) We propose the first mixed-signal mixed-kernel SVM design in FE, implementing RBF kernels in the analog domain and linear kernels in digital.
2) We introduce a separation-driven exploration that trains mixed-kernel SVMs and systematically allocates classifiers to kernel types, achieving high accuracy.

II. Background

A. Support Vector Machine

SVMs are supervised classifiers—robust to overfitting and well-suited for small datasets—that identify a set of m support vectors to determine an optimal hyperplane in feature space, using the following binary decision function for separating between two classes:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i \, K(\mathbf{x}_i, \mathbf{x}) + b\right), \tag{1}$$

where $\{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ are support vector data with $y_i \in \{-1, +1\}$, $\alpha_i \geq 0$ are the dual coefficients, b is the bias, and $K(\cdot, \cdot)$ is a kernel function [21]. For an RBF kernel:

$$K(\mathbf{w}_i, \mathbf{x}) = \exp\big(-\gamma \|\mathbf{x}_i - \mathbf{x}\|^2\big), \quad \gamma > 0, \tag{2}$$

whereas for linear: $K(\mathbf{x}_i, \mathbf{x}) = \mathbf{x}_i^\top \mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^D$ denotes the input features. For the linear case, the dual and primal formulations coincide [22], yielding a single weight vector:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \, \mathbf{x}_i, \qquad f(\mathbf{x}) = \text{sign}\big(\mathbf{w}^\top \mathbf{x} + b\big). \tag{3}$$

Linear kernels reduce inference to one dot product and bias addition, requiring $O(D)$ multiply-accumulate (MAC) operations with only one weight parameter per input and no dependence on support vectors m at run time, compared to $O(m \cdot D)$ for RBF. In practice, RBF kernels can offer stronger non-linear decision boundaries at the cost of higher computation (for euclidean distance and exponential calculation), whereas linear SVMs trade robustness for a simplified, low-latency datapath. We extend SVMs to K-class problems via the One-vs-One (OvO) strategy. In OvO, each binary classifier discriminates between a pair of classes, outputting either 0 or 1 indicating the winner, with predictions made by majority voting over all outputs. As explained later, this property (i.e., analog-in, digital-out) considerably simplifies the integration of analog kernels in our mixed-signal architecture.

B. Flexible Electronics

Recent advancements in FE involve the design of Flexible Integrated Circuits (FlexICs) with Indium Gallium Zinc Oxide (IGZO) Thin-Film Transistors (TFTs), offering mechanical adaptability and flexibility with cost-effective manufacturing [23]. Unlike conventional silicon devices, IGZO TFTs can be made on lightweight, flexible substrates (e.g., polyimide) using low-temperature lithography and without needing protective packaging. Thus, they substantially lower production costs, fabrication time (from 32 weeks to under 3.5 [24]), and environmental impact (e.g., water consumption, carbon emissions). Despite these benefits, IGZO TFTs fall behind CMOS in performance and feature size (e.g., $600\,\text{nm}$ for FlexICs compared to a few nm in silicon [23]). Moreover, the technology provides only n-type devices, restricting designs to unipolar logic and necessitating resistor-NMOS (R-NMOS) circuits, where a pull-up resistor replaces the pMOS transistor. This directly impacts delay and power consumption, and poses significant challenges for designing complex circuits, such as ML classifiers. Our bespoke mixed-signal flexible SVM circuits combat these inherent challenges, by offloading computationally-demanding RBF kernels in the analog domain, reducing both area and power significantly compared to digital deployment.

C. Related Work

Several SVM hardware implementations have been proposed, mostly relying on CMOS technology. On the digital side, [25] provides a review of SVM classifiers implemented on FPGA platforms. A practical example is given in [26], which demonstrates a low-cost FPGA-based SVM classifier for melanoma detection on a Xilinx Zynq device. In contrast, analog CMOS realizations target ultra-low-power operation, such as the fully trainable Gaussian-kernel SVM presented in [27]. More recently, [28] reported a low-power analog integrated implementation of SVM with on-chip learning, tested on a bearing fault detection application. However, these prior works rely on rigid silicon-based technologies, which lack mechanical properties such as flexibility and conformability, and also do not face the inherent limitations of FE.

In the context of printed and flexible electronics [29], SVM classifiers have been investigated primarily along two directions: (i) fully-parallel approximated architectures that reduce hardware cost at the expense of accuracy [11], [12], and (ii) sequential architectures that significantly lower area but incur high energy consumption [30], [31]. In [11], hardware-friendly weight replacement and gate-level pruning were proposed as post-training approximations, while [12] extended this with voltage over-scaling at the circuit level. More recently, [30], [31] further explored sequential SVM architectures, emphasizing the trade-off between area and energy consumption. However, these implementations only consider linear kernels, avoiding the hardware cost and design complexity of digital RBF kernels but resulting in diminished accuracy—even more so in approaches that exploit approximate computing.

III. Proposed Mixed-signal Mixed-kernel SVMs

A. Architecture Overview

Fig. 1 illustrates an example of our proposed mixed-signal mixed-kernel SVM architecture for three classes, which partitions linear and RBF kernel across digital and analog domains, respectively. The linear classifiers require analog-to-digital converters (ADCs) to digitize their inputs, whereas the analog RBF classifiers operate directly on analog signals. Per the OvO scheme, one binary classifier is instantiated for each two classes, yielding a single binary value (0 or 1, according to the winning class) for both RBF and linear classifiers. Thus, as analog classifiers produce digital outputs,

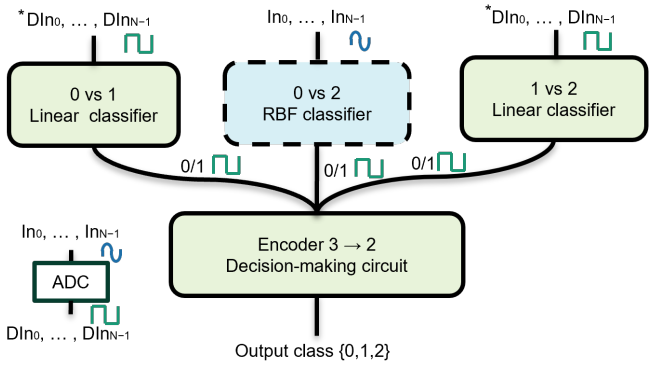


Fig. 1: Overview of the proposed mixed-kernel, mixed-signal SVM for three classes: RBF classifiers are realized in analog, while linear classifiers and decision logic are in digital.

no additional conversion is needed, eliminating potential ADC costs. This also considerably simplifies the result aggregation by the digital decision-making circuitry, responsible for encoding the predicted class. All classifiers are realized in parallel, eliminating the need for control circuitry or sequential elements—costly in FE.

B. Analog RBF Classifier

First, we describe the design of our analog RBF classifier. Designing in FlexIC presents challenges compared to CMOS, as the technology includes only n-type transistors and lacks p-type devices. Still, based on the CMOS kernel design of [21], we implement the RBF classifier in our flexible technology as shown in Fig. 2, using an n-type transistor differential pair biased in subthreshold, leveraging the exponential I–V characteristic of this regime. Subthreshold biasing enables ultra-low power, aligning with our target applications. Even though the reduced bias currents may introduce longer settling time, this latency is effectively hidden in the mixed-signal design by the ADC conversion time in the digital path, providing timing slack for the analog stage. Also, FE circuits typically operate within the Hz range, since performance is a lower priority to area and power efficiency.

1) RBF kernel (Gaussian): The kernel is implemented with two subthreshold n-type transistors differential pairs, (Q_1, Q_2) cascaded by (Q_3, Q_4), and Q_5 for biasing the kernel and setting the operating point. The circuit produces a current-mode bell-shaped response that, near the origin, is well modeled by a squared hyperbolic secant. Specifically, cascading the complementary pairs gives, with $x = \frac{\Delta v}{nV_T}$:

$$I_{\text{out}} = \frac{I_{\text{in}}}{\left(1+e^{-x}\right)\left(1+e^{x}\right)} = \frac{I_{\text{in}}}{4}\,\operatorname{sech}^2\!\left(\frac{x}{2}\right), \qquad (4)$$

where n is the subthreshold slope factor, V_T is the thermal voltage and $\Delta v = V_1 - V_2$. Following a Taylor expansion about $\Delta v = 0$, we have $\operatorname{sech}^2 u = 1 - u^2 + O(u^4)$ with $u = \frac{\Delta v}{2nV_T}$, while $\exp(-\gamma\,\Delta v^2) = 1 - \gamma\,\Delta v^2 + O(\Delta v^4)$. Matching

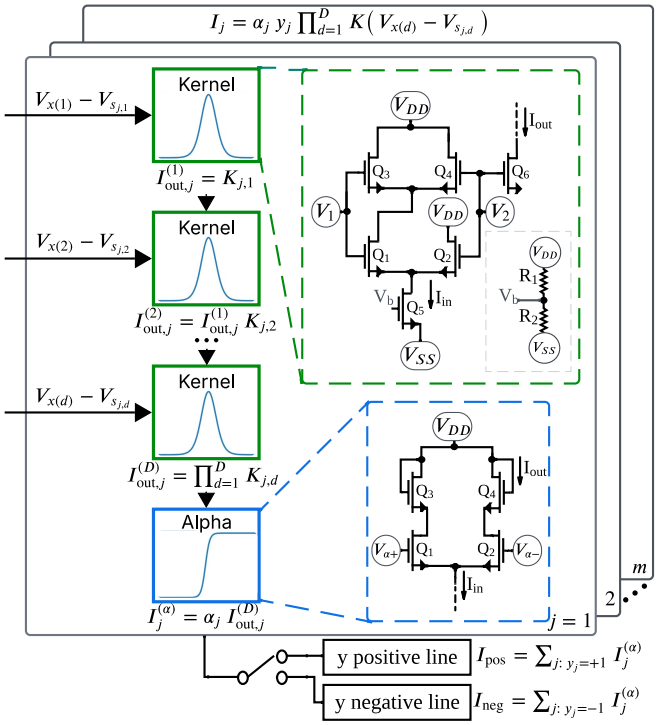


Fig. 2: Analog RBF classifier architecture: a hardware Gaussian kernel forms a separable product across dimensions; outputs are weighted by α and routed by the sign y, then differentially accumulated across support vectors.

the quadratic terms gives $\gamma = \frac{1}{4n^2V_T^2}$, so near the origin, the cascaded pair provides a Gaussian approximation to the kernel:

$$I_{\text{out}}(\Delta v) \;\approx\; \frac{I_{\text{in}}}{4}\,\exp\!\Big(-\gamma\,\Delta v^2\Big), \qquad (5)$$

where σ denotes the standard deviation. Equivalently, we use $\gamma = 1/2\sigma^2 \approx 1/(8n^2V_T^2)$, which aligns with γ from (2).

The output current is sensed by an n-type transistor Q_6 whose gate and source are tied to Q_4. In subthreshold, the drain current depends exponentially on V_{GS} and only weakly on V_{DS}, so Q_5 produces a faithful, scaled replica of Q_4's current. Because we operate in subthreshold—where I_D is only weakly dependent on V_{DS}—we use device ratioing for readout. Therefore, we undersize Q_4 (small W/L) to minimize loading, and set Q_6 with a W/L similar to Q_3 so that the readout branch provides a larger copy of the kernel current for the next stage.

2) Kernel Product Across Dimensions: To handle D dimensions, we implement a separable kernel by cascading D identical one-dimensional cells per support vector. For support vector j and dimension d, we form $\Delta v_{j,d} = V_{x(d)} - V_{s_{j,d}}$ and apply (5) to that stage. Because the output of stage $(d-1)$ drives the input of stage d ($I^{(d)}_{\text{in},j} = I^{(d-1)}_{\text{out},j}$), the exponential factors multiply across stages. After D stages, for

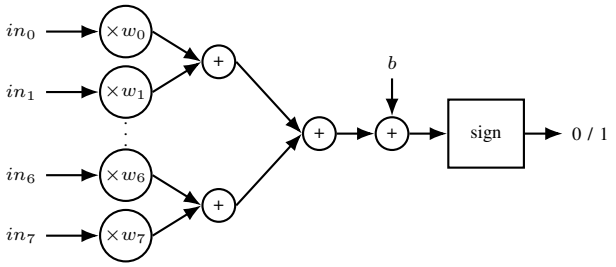


Fig. 3: Datapath of an 8-input linear SVM: weighted inputs are summed with bias, and the sign determines the output.

each $j = 1, \ldots, m$ we obtain:

$$I_{\text{out},j}^{(D)} = \prod_{d=1}^{D} K_{j,d} = \frac{I_{\text{in},j}}{4^D} \exp\Big(-\gamma \sum_{d=1}^{D} \Delta v_{j,d}^2 \Big). \qquad (6)$$

When a large number of features is employed, the summation chain grows, resulting in current degradation that can no longer be reliably distinguished. Because the FE process is n-type transistors-only (no p-type devices), accurate current mirroring from n-type outputs is challenging. Therefore, we support up to five inputs in our design.

3) Alpha Multiplier: This block scales the kernel current by the dual coefficient through a controllable factor $\alpha \in (0, 1)$. As shown in Fig 2, it is implemented with a subthreshold differential pair (Q_1, Q_2) with diode-connected loads (Q_3, Q_4). With control differential $\Delta V_\alpha = V_{\alpha+} - V_{\alpha-}$, the branch current follows a logistic function:

$$I_j^{(\alpha)} \;=\; I_{\text{out},j}^{(D)} \frac{1}{1 + \exp\big(\frac{\Delta V_\alpha}{n V_T}\big)} \;\equiv\; I_{\text{out},j}^{(D)} \alpha(\Delta V_\alpha)\,.$$

4) Signed Accumulation over Support Vectors: As mentioned in (1), each support vector j produces a nonnegative current $I_j = \alpha_j K_j(\mathbf{x})$. A switch controlled by the label $y_j \in \{+1, -1\}$, routes I_j to the $+$ rail when $y_j = +1$ or to the $-$ rail when $y_j = -1$. The two rails passively sum their incoming currents; Thereby realizing negative weights by accumulation on the negative rail.

Finally, the classifier's score is passed through a comparator to obtain its sign, provides a digital output and seamlessly feeds the digital decision making circuit without requiring any additional digitization, i.e., without an ADC.

C. Digital Linear Classifier & Decision-Making Logic

In the digital domain, we implement the linear binary classifiers together with the multiclass decision-making logic of our mixed-signal SVM architecture. Similar to state-of-the-art flexible SVM designs [9], our linear kernels are designed as bespoke fully-parallel architectures, essentially implementing fixed-point MAC operations, as shown in Fig. 3. Trained weights are quantized and multiplied with the quantized features as obtained from the ADC. All products are computed in parallel and accumulated through an adder tree, followed by bias addition and sign evaluation. This yields a memory-less architecture, eliminating the considerable overhead of sequential elements. Note, non-volatile memories remain scarce and costly in FE [23]. In addition, weight and bias coefficients are hardwired within each linear classifier, following the bespoke design paradigm [23], substantially improving the area and power of flexible multipliers and adders compared to conventional (i.e., non-customized) implementations.

The outputs of both digital linear and analog RBF classifiers are aggregated by the digital decision-making circuit to perform multiclass classification. Instead of majority voting as in [9], [12], decision-making is realized with an encoder, as shown in Fig. 1. Each binary classifier provides a one-bit output, and the encoder maps the collection of these binary outcomes directly to the corresponding class label. This hardware-efficient scheme removes the need for dedicated counter and `argmax` circuitry, thereby enabling the seamless integration of both digital and analog SVM components with minimal area and power overhead.

IV. High-Level Mixed-Kernel Exploration

A. Analog RBF Modeling

For seamless integration with digital simulations—without resorting to full mixed-signal co-simulation, which is complex and requires excessive circuit runs- we develop a high-level behavioral model of the analog RBF core. The model captures the Gaussian kernel and the α multiplier via parameters identified from targeted DC SPICE sweeps, enabling system-level accuracy evaluation. This approach increases automation at a high level while preserving final accuracy through calibration against measured curves. Below, the modeling of each component in the analog RBF classifier (see Section III-B and Fig. 2) is described.

1) Gaussian kernel Modeling: We model the hardware kernel by running a DC analysis, sweeping the differential input Δv, and using the resulting samples $I_{\text{out}}(\Delta v)$. However, in order to measure the γ of the simulated design, we fit an *ideal Gaussian* to the measured transfer characteristic—specifically estimating A_0, γ_0, and μ:

$$I_{\text{out}}(\Delta v) \;\approx\; A_0 \exp\big[-\gamma_0 \, (\Delta v - \mu)^2 \big]. \qquad (7)$$

We then use the SPICE data together with the fitted γ_0 in the behavioral modeling.

We fix the operating point at a specific bias V_b. To achieve different RBF widths γ^*, we apply input scaling relative to the measured γ_0 and the separable kernel is then evaluated as:

$$s_\gamma = \sqrt{\gamma^*/\gamma_0}, \qquad K(\Delta v) = \prod_{d=1}^{D} K(s_\gamma \, \Delta v). \qquad (8)$$

2) Kernel product across dimensions: We multiply the per-dimension kernels to realize a separable D-dimensional kernel. No additional approximation is introduced here: the current is passed from one stage to the next; only the input scaling s_γ differs.

3) Alpha multiplier: We model the dual-coefficient magnitude $\alpha \in (0, 1)$ with a logistic curve, fitted to SPICE by a DC sweep of the control differential ΔV_α versus the measured

Algorithm 1 Mixed-kernel SVM selection process.

Require: Labeled dataset $\mathcal{D}$ with classes $\{c_1, \ldots, c_K\}$
$\mathcal{P} \leftarrow \{(c_i, c_j) \mid i < j\}$ ▷ All OvO pairs
$\mathcal{K} \leftarrow \emptyset$ ▷ Chosen kernel per pair
$\mathcal{C} \leftarrow \emptyset$ ▷ Trained classifier per pair
for all $(c_i, c_j) \in \mathcal{P}$ **do**
  $\mathcal{D}_{ij} \leftarrow \{(x, y) \in \mathcal{D} \mid y \in \{c_i, c_j\}\}$
  Train linear C_{ij}^{lin} and RBF C_{ij}^{rbf} on $\mathcal{D}_{ij}$
  $A_{\text{lin}} \leftarrow \text{acc}(C_{ij}^{\text{lin}})$, $A_{\text{rbf}} \leftarrow \text{acc}(C_{ij}^{\text{rbf}})$
  if $A_{\text{rbf}} > A_{\text{lin}}$ **then**
    $k_{ij} \leftarrow$ RBF, $C_{ij} \leftarrow C_{ij}^{\text{rbf}}$
  else
    $k_{ij} \leftarrow$ Linear, $C_{ij} \leftarrow C_{ij}^{\text{lin}}$
  end if
  $\mathcal{K} \leftarrow \mathcal{K} \cup \{k_{ij}\}$, $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_{ij}\}$
end for
Decision step (encoder): For input x, compute binary outputs $b_{ij}(x) \in \{0, 1\}$ from all $C_{ij} \in \mathcal{C}$; obtain $\hat{y}$
Evaluation: $\text{Acc}(\text{SVM}_{\text{mixed}}) \leftarrow \text{acc}(\hat{y}; \mathcal{D})$
Outputs: $\mathcal{K}$ (kernel map), $\mathcal{C}$ (mixed-kernel classifiers)

ratio $\alpha = I_{\text{out}}/I_{\text{in}}$. The fit yields (x_0, s), and in software we map a desired α to its control value via:

$$\Delta V_\alpha = x_0 + s \ln\left(\frac{1}{\alpha} - 1\right) \tag{9}$$

B. Mixed-Kernel Mapping & Training

Leveraging that SVM OvO multi-class classification decomposes the final classification to distinct binary classification problems, we propose a separation strategy to assign each binary classifier to the optimal kernel type, RBF or linear. Our objective is to maximize the accuracy of each binary classifier—enabling superior model-level accuracy compared to single-kernel state-of-the-art approaches—whilst minimizing the number of RBF kernels.

Algorithm 1 summarizes our mapping strategy. For a K-class task, we enumerate all $\binom{K}{2}$ pairs of classes (c_i, c_j) and extract the corresponding binary subset for each pair. Each pair is first trained with a linear classifier, and its accuracy is used to gauge classification difficulty. If the accuracy is sufficient, the pair remains linear; if not, it is reassigned to an RBF kernel. In this way, most classifiers remain lightweight and digital, while only the challenging ones are realized as analog RBFs. This selective allocation preserves the accuracy benefits of RBFs where they matter most, while minimizing their overhead and maintaining an overall efficient design. Finally, after all classifiers are extracted, they are integrated within a unified SVM model, alongside the decision-making logic for outputting the predicted class, per the OvO scheme. Floating-point RBF classifiers are replaced with our hardware-accurate high-level analog model of Section IV-A. The exploration therefore outputs: (i) a pairwise kernel assignment map distinguishing linear (digital) from RBF (analog) and (ii) the mixed-kernel SVM model, ready for system-level inference.

V. Results & Analysis

A. Experimental Setup

1) Software Setup: We evaluate our mixed-kernel SVM approach over 3 datasets from the UCI ML repository [32], as they are well-suited for sensor-based FE applications [9]. Datasets include Balance Scale (Balance/Bal.), Seeds and Vertebral 3 Columns (Vertebral/V3C). Sensor data are normalized within [0,1], and any non-sensor (categorical) features are removed during pre-processing. We split each dataset into 70% training and 30% testing. Feature selection is applied before training to align the input dimensions with our analog hardware constraints (see Section III-B2), limiting up to 5 input features. Scikit-learn is used for SVM training and feature selection.

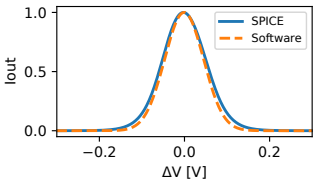
2) Hardware Setup: Sensory inputs are uniformly quantized to 4-bit fixed-point precision. For linear SVMs, weights and biases are quantized following [12] to preserve accuracy. For digital RBF SVMs support vectors and dual coefficients are quantized to ensure sufficient precision. For analog simulations and digital circuit synthesis and simulation, Cadence Spectre simulator, Synopsys Design Compiler T-2021.06 and VCS T-2022.03 are used, respectively. Our designs are mapped to PragmatIC's Gen3 FlexIC 1.0.0 PDK at 1.5 V [33]. The supply voltage for the analog is set to 1 V, regulated down from the digital supply voltage. The dimensions of all components in our analog RBF classifier design are reported in Table I, for the comparator, we estimated power and area from the comparator designed in [34]. Both analog and digital target performance at 2 Hz. It should be noted that the target applications of FE are inherently low-throughput, and therefore the chosen operating frequency aligns with the typical speed requirements of such systems [23].

TABLE I: Component dimensions of analog RBF classifier

Gaussian Kernel	
Q_1–Q_3, Q_6	W = 40 µm, L = 0.6 µm
Q_4	W = 1 µm, L = 0.6 µm
Q_5	W = 20 µm, L = 1.2 µm
R_1= 10 MΩ	W = 0.6 µm, L = 28.5 µm
R_2 = 4.28 MΩ	W = 0.6 µm, L = 12.2 µm
Alpha Multiplier	
Q_1–Q_4	W = 40 µm, L = 0.6 µm

Component	nRMSE	r
Gaussian kernel (at $V_b = 0.30$ V)	0.0218	0.997
Product across dims ($D = 3$)	0.0117	0.998
Alpha multiplier (logistic fit)	0.0003	0.999

(a) SPICE vs. ideal reference



(b) Gaussian kernel fit vs. SPICE

Fig. 4: Analog RBF validation: (a) SPICE vs. ideal metrics and (b) Gaussian kernel comparison.

B. Analog RBF validation

First, we validate the behavior of our analog RBF classifier design, and its comprising components. Specifically, we compare the obtained SPICE simulation traces against an *ideal* software reference for the following components: the Gaussian kernel cell, the product across dimensions, and the alpha-multiplier. For each case, we report the normalized

TABLE II: Evaluation against single-kernel state-of-the-art.

Dataset	Design	Circuit Acc. (%)	Total Area (mm^2)	Total Power (mW)	RBF/linear classifier ratio
Bal.	Linear (digital)	92	0.024	0.004	0/3
	RBF (digital)	93	13.400	2.230	3/0
	Ours (mixed)	92	0.062	0.081	1/2
Seeds	Linear (digital)	92	0.067	0.011	0/3
	RBF (digital)	95	7.000	1.190	3/0
	Ours (mixed)	95	0.125	0.092	1/2
V3C	Linear (digital)	69	0.092	0.014	0/3
	RBF (digital)	83	5.600	0.960	3/0
	Ours (mixed)	89	0.108	0.088	2/1

root mean square error (nRMSE) and the correlation between SPICE and the ideal outputs. The results are summarized in Fig. 4(a). Overall, we observe excellent agreement with the theoretical reference, with all blocks exhibiting very low nRMSE and near-unity correlation ($r \approx 1$). Fig. 4(b) also presents the I-V curve of the Gaussian kernel in comparison to the ideal software function, providing a visual confirmation of this alignment. Thus, our analog implementation provides accurate (and well-correlated) outputs, faithful to the ideal RBF implementation outlined in Section III-B.

C. System Evaluation

Next, we evaluate our mixed-signal and mixed-kernel SVMs in terms of accuracy and hardware efficiency. Specifically, we focus our analysis on the following: (i) comparison of single–kernel baselines (digital linear, digital RBF, and our mixed-signal mixed-kernel SVM), (ii) analysis of analog vs. digital RBF efficiency, (iii) a breakdown of linear and RBF contributions in the mixed design to quantify area and power.

We compare our SVMs against the state-of-the-art SVM designs, which focus solely on single–kernel linear digital implementations. In addition, we design against single-kernel digital RBF SVMs, aiming for more comprehensive comparisons. Table II reports the results in terms of circuit accuracy, total area, and power. Compared to purely-linear SVMs, our mixed-kernel designs offer superior classification accuracy, with an average increase of 7.7%, reaching up to 20% for Vertebral. This comes at the cost of moderate overhead, since digital linear classifiers yield lower area and power cost compared to analog RBF ones—by 2.5x and 12.4x, respectively. In contrast, compared to single-kernel digital RBF SVMs, our designs offer orders-of-magnitude higher efficiency, with an average area and power gain of 108x and 17×, respectively. At the same time, they achieve comparable accuracy, ranging from a minor 1% drop for Balance, to equal values for Seeds, up to a 6% gain for Vertebral, showcasing the advantages of mixed-kernel SVMs. *Overall, our mixed–kernel and mixed-signal SVMs offer the best accuracy–efficiency trade–off, surpassing state-of-the-art accuracy of linear SVMs while providing huge hardware gains over digital RBF designs.*

Note, we do not provide a direct comparison against single-kernel RBF-based SVMs implemented purely in the analog domain, since we are the first to design analog RBF classifiers in FlexIC technology. In addition, our mixed-signal designs deliver more favorable trade-offs given the hardware constraints of FE applications. We achieve high accuracy—within 1% of the software accuracy—while the analog RBF classifier is costlier than its digital linear counterpart.

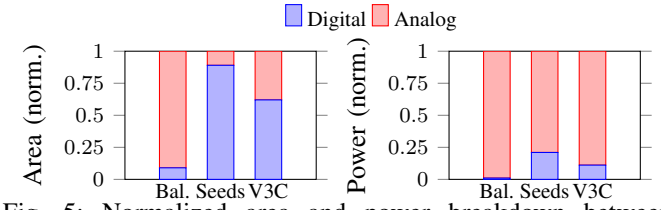


Fig. 5: Normalized area and power breakdown between analog/digital in our mixed-signal SVMs of Table II.

Next, we perform a direct comparison between digital RBF classifiers from (Table II) against our analog designs. Our analog classifiers are on average 109× more area-efficient and 16× more power-efficient, highlighting the importance of analog realization, and enabling for the first time RBF-based SVMs in FlexIC technology.

Finally, we present an area and power breakdown of our mixed-kernel solutions of Table II between their analog and digital components. Fig. 5 presents the normalized results per the studied datasets. Overall, the digital end (i.e., linear classifiers and decision making) accounts for 54% of the area, mainly due to the separation nature of our exploration, which aims to minimize the count of RBF blocks. Interestingly, the analog RBF dominates the total area for Balance by nearly 90%. This occurs because the digital linear component converged to zero or power of 2 weights, which in this bespoke architecture directly translates to hardware savings both area and power (99% static power in FE [23] by removing the corresponding multipliers. According to Fig. 5(b), the analog RBF classifiers dominate the total power consumption by 89%, on average. This reflects the inherently higher power overhead of analog computation in FE, despite its area efficiency.

VI. Conclusion

Flexible electronics (FE) are emerging as a pathway to low-cost, conformal computing for sensor-driven applications. However, implementing ML algorithms in FE remains costly due to the limited efficiency of complex circuits. In this work, we propose a mixed-kernel SVM that combines digital linear kernels with analog RBF. Our evaluations highlight that our mixed approach achieves, on average 7.7% higher accuracy than linear baselines with minimal hardware overhead, while reducing area and power by up to 108× and 17×, respectively, compared to fully digital RBFs. These results demonstrate that analog RBFs, when coupled with a lightweight digital pipeline, provide a practical and scalable route to hardware-efficient intelligence in FE systems.

Acknowledgements

This work is partially supported by the European Research Council (ERC) (Grant No. 101052764), and co-funded by the H.F.R.I call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" (H.F.R.I. Project Number: 17048).

REFERENCES

[1] S. Xu, Y. Huang, and J. A. Rogers, "Epidermal electronics," *Science*, vol. 344, no. 6179, pp. 70–72, 2014.

[2] J.-W. Lee *et al.*, "High sensitivity flexible paper temperature sensor and body-attachable patch for thermometers," *Sensors and Actuators A: Physical*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924424720305495

[3] Y. Li *et al.*, "The soft-strain effect enabled high-performance flexible pressure sensor and its application in monitoring pulse waves," *Research*, 2022.

[4] J. Gao *et al.*, "Ultra-robust and extensible fibrous mechanical sensors for wearable smart healthcare," *Advanced Materials*, vol. 34, no. 20, p. 2107511, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.202107511

[5] S. Agarwala *et al.*, "Wearable bandage-based strain sensor for home healthcare: Combining 3d aerosol jet printing and laser sintering," *ACS Sensors*, vol. 4, no. 1, pp. 218–226, 2019.

[6] K. Zhou, R. Ding, X. Ma, and Y. Lin, "Printable and flexible integrated sensing systems for wireless healthcare," *Nanoscale*, vol. 16, pp. 7264–7286, 2024. [Online]. Available: http://dx.doi.org/10.1039/D3NR06099C

[7] X. Luo, "Application of inkjet-printing technology in developing indicators/sensors for intelligent packaging systems," *Current Opinion in Food Science*, vol. 46, p. 100868, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214799322000704

[8] A. Beniwal, P. Ganguly, A. K. Aliyana, G. Khandelwal, and R. Dahiya, "Screen-printed graphene-carbon ink based disposable humidity sensor with wireless communication," *Sensors and Actuators B: Chemical*, vol. 374, p. 132731, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925400522013740

[9] M. H. Mubarik *et al.*, "Printed machine learning classifiers," in *Annu. Int. Symp. Microarchitecture (MICRO)*, 2020, pp. 73–87.

[10] J. Henkel *et al.*, "Approximate computing and the efficient machine learning expedition," in *International Conference On Computer Aided Design (ICCAD)*, 2022, pp. 1–9.

[11] 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.

[12] G. Armeniakos, G. Zervakis, D. Soudris, M. B. Tahoori, and J. Henkel, "Model-to-circuit cross-approximation for printed machine learning classifiers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2023.

[13] 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 Transactions on Computers*, pp. 1–8, 2023.

[14] F. Afentaki *et al.*, "Bespoke approximation of multiplication-accumulation and activation targeting printed multilayer perceptrons," in *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, 2023, pp. 1–9.

[15] F. Afentaki, M. Hefenbrock, G. Zervakis, and M. B. Tahoori, "Embedding hardware approximations in discrete genetic-based training for printed mlps," in *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2024, pp. 1–6.

[16] V. Mrazek *et al.*, "Evolutionary approximation of ternary neurons for on-sensor printed neural networks," in *International Conference On Computer Aided Design (ICCAD)*, 2024.

[17] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[18] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[19] J. R. Cavallaro, H.-J. Lee, and K. Yang, "Analog circuit implementations of kernel-based svm classifiers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 6, pp. 1441–1454, 2010.

[20] X. Li, H. Jiang, R. Shafik, and et al., "Low-power analog implementation of radial basis function kernel for svm," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.

[21] K. Kang and T. Shibata, "An on-chip-trainable gaussian-kernel analog support vector machine," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1513–1524, 2010.

[22] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

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

[24] PragmatIC Semiconductor, "Sustainability through flexible electronics," https://www.pragmaticsemi.com/sustainability/, 2023, whitepaper.

[25] S. Afifi, H. GholamHosseini, and R. Sinha, "Fpga implementations of svm classifiers: A review," *SN computer science*, vol. 1, no. 3, p. 133, 2020.

[26] S. Afifi, H. GholamHosseini, and R. Sinha, "A low-cost fpga-based svm classifier for melanoma detection," in *2016 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, 2016, pp. 631–636.

[27] K. Kang and T. Shibata, "Analog gaussian-kernel support vector machine: A fully trainable vlsi approach," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1513–1524, 2009.

[28] V. Alimisis, G. Gennis, M. Gourdouparis, C. Dimas, and P. P. Sotiriadis, "A low-power analog integrated implementation of the support vector machine algorithm with on-chip learning tested on a bearing fault application," *Sensors*, vol. 23, no. 8, p. 3978, 2023.

[29] M. B. Tahoori, E. Ozer, G. Zervakis, K. Balaskas, and P. Pal, "Computing with printed and flexible electronics," in *2025 IEEE European Test Symposium (ETS)*. IEEE, 2025, pp. 1–9.

[30] I. Sertaridis, S. Besias, F. Afentaki, K. Balaskas, and G. Zervakis, "Compact yet highly accurate printed classifiers using sequential support vector machine circuits," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2025.

[31] S. Besias, I. Sertaridis, F. Afentaki, K. Balaskas, and G. Zervakis, "Late breaking results: Energy-efficient printed machine learning classifiers with sequential svms," 2025.

[32] D. Dua and C. Graff, "UCI machine learning repository," 2017.

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

[34] M. Shatta *et al.*, "Invited paper: Feature-to-classifier co-design for mixed-signal smart flexible wearables for healthcare at the extreme edge," 2025. [Online]. Available: https://arxiv.org/abs/2508.19637