To appear at the 2025 IEEE International Symposium on Circuits and Systems (ISCAS), May 25-28 2025, London, UK.

Compact Yet Highly Accurate Printed Classifiers Using Sequential Support Vector Machine Circuits

Ilias Sertaridis, Spyridon Besias, Florentia Afentaki, Konstantinos Balaskas and Georgios Zervakis
Department of Computer Engineering & Informatics, University of Patras, Patras, Greece
{st1072480, st1072524, afentaki, kompalas, zervakis}@ceid.upatras.gr

***Abstract*—Printed Electronics (PE) technology has emerged as a promising alternative to silicon-based computing. It offers attractive properties such as on-demand ultra-low-cost fabrication, mechanical flexibility, and conformality. However, PE are governed by large feature sizes, prohibiting the realization of complex printed Machine Learning (ML) classifiers. Leveraging PE's ultra-low non-recurring engineering and fabrication costs, designers can fully customize hardware to a specific ML model and dataset, significantly reducing circuit complexity. Despite significant advancements, state-of-the-art solutions achieve area efficiency at the expense of considerable accuracy loss. Our work mitigates this by designing area- and power-efficient printed ML classifiers with little to no accuracy degradation. Specifically, we introduce the first sequential Support Vector Machine (SVM) classifiers, exploiting the hardware efficiency of bespoke control and storage units and a single Multiply-Accumulate compute engine. Our SVMs yield on average 6x lower area and 4.6% higher accuracy compared to the printed state of the art.**

***Index Terms*—Machine Learning, Support Vector Machines, Printed Electronics**

I. Introduction

Printed Electronics (PE) have emerged as a complement to silicon-based technology, meeting demands that are untouchable by the latter, such as mechanical flexibility, non-toxicity, conformability, and ultra-low cost [4]. Target applications include smart packaging [5], [6], forensics [7], and accessible healthcare products and wearables [8]–[12]. Unlike conventional silicon technologies, which are constrained by high manufacturing and assembly costs, PE leverage additive, mask-less printing processes that enable on-demand production of flexible circuits at significantly lower costs [13].

Such application domains stand to greatly benefit from the infiltration of Machine Learning (ML) algorithms [1]. However, PE face significant challenges in implementing complex circuits such as ML classifiers, due to the large feature sizes and the corresponding power and area overheads, thus hindering the ubiquitous integration of ML in PE applications. To overcome these limitations, leveraging the low non-recurring engineering (NRE) and fabrication costs in PE, bespoke circuits have been proposed as a promising solution [1]. They refer to fully customized designs tailored to a specific ML model and dataset, optimizing hardware for a particular application. Such tailored designs allow for significant reductions in power and area, by hardwiring the parameters of the ML model into the circuit implementation. This level of customization is infeasible in conventional lithography-based silicon technologies, due to elevated costs (e.g., NRE, maskset, etc.), particularly at low to moderate volumes; even FPGA-based systems are constrained by the architecture, multiplexing, and routing of the underlying fabric. Additionally, exploiting the intrinsic error resilience of ML circuits [14], state-of-the-art approaches combine bespoke circuits with approximate computing to boost hardware efficiency [2], [3], [15]–[18]. However, they often lead to significant accuracy loss in order to meet the tight power and area constraints inherent in PE.

A wide range of digital printed ML classifiers have been showcased in the literature, with the main focus lying on fully parallel implementations of shallow Neural Networks [3], [16]–[18]. In this work, we leverage the high accuracy of Support Vector Machines (SVMs) in target applications and propose, for the first time, *printed sequential SVM classifiers*. Our goal is to minimize area while maintaining high accuracy. We design a compute engine with just one Multiply-Accumulate (MAC), folding each support vector computation over it, maximizing hardware reuse and significantly reducing area. Additionally, we reduce the area requirements associated with sequential elements by minimizing the use of registers–which are costly in PE–and we use bespoke multiplexers (MUXs) for storing model parameters. A bespoke control unit further folds the entire SVM computation over the single support vector engine. We implement the One-vs-One (OvO) algorithm [19] into a binary decision tree, simplifying the control circuitry and eliminating the need for a voter and additional registers. Compared to printed parallel SVMs, our sequential ones achieve more than 5x lower area at similar accuracy. **Our novel contributions within this work are as follows**:

1) We propose, design, and evaluate, for the first time, printed sequential SVM circuits targeting printed ML classification. The hardware description of our SVMs is automatically generated[1].
2) Our SVMs constitute the most accurate digital printed ML classifiers that feature acceptable area and power overheads, and also meet the strict physical constraints and limited power sources of PE.

II. Related Work

In recent years, research on printed ML classifiers has proliferated, with a primary focus on improving hardware efficiency while adhering to the strict area and power budget of PE systems. As a result, classification accuracy is often compromised in favor of design feasibility. The authors in [1] introduced printed ML classifiers by exploring various models

[1] Our circuits are available on: https://github.com/floAfentaki/Support-Vector-Machine-Circuits-Targeting-Printed-Electronics

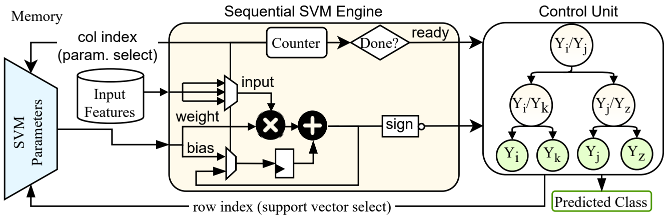

Fig. 1. Overview of our proposed sequential SVM circuits. They consist of three main components memory, SVM compute engine and control unit.

and architectures, and concluding that only simpler models like decision trees and SVM regressors could be implemented in PE. Furthermore, [1] determined that fully parallel architectures should be used in PE. Since then, significant research efforts have focused on fully parallel approximate neural networks (Multi-Layer Perceptrons (MLPs) [2], [3], [15]–[17] and Binary Neural Networks (BNNs) [18]) and also approximate SVMs classifiers [2], [15]. While works like [16]–[18] achieve impressive area efficiency, they often report accuracy losses around 4%-5% or more, due to employing aggressive power-of-two or ternary quantization and approximate additions. On the other hand, [15] introduced a more conservative post-training approximation technique involving hardware-friendly weight replacement to approximate the multiplications in SVMs and MLPs. [15] also applied a gate-level pruning approximation. The authors in [2] expand upon [15] by additionally incorporating voltage over-scaling. In-training hardware-friendly weight replacement along with truncated addition is proposed in [3] for MLPs. Similar to the aforementioned works, we consider the Electrolyte-Gated FET (EGFET) technology [4], which offers good mobility characteristics and operation at low supply voltages (0.6V-1V [20]), aligning well with printed battery-powered applications.

Our work distinguishes from existing approaches by introducing the first design of printed sequential SVM classifiers, demonstrating for the first time that digital printed classifiers can meet the physical constraints of PE with high accuracy.

III. Printed Bespoke Sequential SVMs

SVM is a supervised learning algorithm able to classify data by determining the optimal hyperplane for separating classes in a high-dimensional feature space. SVMs are effective in handling small, high-dimensional datasets, offering robustness against overfitting. In brief, SVMs compute a number of support vectors and based on the obtained results, determine the class with the most wins. In printed hardware, linear kernels are typically preferred within support vectors, to maximize area efficiency. Existing implementations [1], [2] design fully parallel bespoke circuits, where model parameters are hardwired into the circuit implementation, eliminating the need for costly sequential elements in PE, but requiring a hardware multiplier for each SVM weight. In contrast, our work focuses on designing sequential SVMs to minimize the required arithmetic units, making it crucial, nevertheless, to optimize the cost of memory and registers. Notably, in CMOS technologies, a D Flip-flop's area-equivalent is 4 NAND gates, whereas in EGFET, this number rises to 6–a 50% increase.

An abstract overview of our proposed sequential SVM circuit is shown in Fig. 1. It consists of three main components, i.e., parameter storage, control, and compute, all working in sync to fold the entire SVM prediction over a single MAC unit. Each iteration involves fetching the respective support vector from memory (as determined by the control unit) and computing it in the support vector engine (a multi-cycle operation over a single MAC). The result is fed back to the control unit, which either advances the computation (intermediate step) or selects the winning class (final step). Importantly, our design choices for implementing our sequential SVM circuits target area efficiency. Nevertheless, since power consumption in EGFET circuits is mostly static and internal (short-circuit), minimizing area also effectively reduces power.

A. SVM Training

Two established strategies for SVM multi-class classification are OvO and OvA (One-vs-All) [21]. In OvO, each binary classifier distinguishes between two classes, requiring $\frac{n(n-1)}{2}$ classifiers for n classes. OvA uses n classifiers, each separating one class from the rest. OvO requires storing more support vectors but works with smaller subsets of training data, avoiding accuracy loss from imbalanced datasets. Prioritizing high accuracy and relying on the inherent area efficiency of our sequential implementation, we choose OvO. For the examined datasets (Section IV), using quantized support vectors, OvO delivers 8.7% higher accuracy, on average, compared to OvA.

Training is performed using scikit-learn's LinearSVC class to train a classifier (support vector) for each possible class pair. Inputs are truncated to 4-bit fixed-point, and post-training, weights and biases are quantized with min-max linear scaling to the lowest precision that results in negligible accuracy loss (within 0.5%). Additionally, SVM inference is profiled to determine the minimum precision needed for the support vector engine's accumulator. The Verilog description for each trained SVM is automatically generated using code templates. The extracted precisions and SVM parameters are stored in a configuration file. The support vector engine is the same, with only precision adjustments, while a unique control unit is generated for each model.

B. Support Vector Storage

Sequential architectures must store model parameters in memory (weights and biases in our SVMs). A printed Read-Only Memory (ROM) is a prominent choice for this. A compact ROM was proposed and fabricated in [4], outperforming other state-of-the-art designs. It uses a crossbar architecture, where cross-points are shorted by printing conductive material (such as PEDOT:PSS) to represent bit values. By varying the geometry of the conductive material, multi-bit values can be stored in a single printed dot. An analog-to-digital converter (ADC) is needed to read stored values. Considering the cost of ADCs and ROM cells in [4], we deduce that storing 2-bit values is optimal for reading 2-bit to 8-bit words as for our model parameters. This assumes the use of one to four 2-bit ADCs to maintain reasonable performance. Since our design uses a single-MAC SVM engine, we need to process only a single model parameter (weight or bias) per cycle. One support vector is stored per crossbar row. Based on the selected support vector (from the control unit), the respective row is activated. Then, a set of columns, depending on the precision of the model parameters, are activated each cycle. A counter (see Section III-C) that selects the appropriate parameter is used as the column index. Nevertheless, such an architecture is highly vulnerable to printing variations, which can alter stored parameters and lead to significant accuracy degradation [22].

As a more robust alternative, we propose the use of bespoke MUXs. The MUX input data are hardwired to the model parameters, with the row and column indexes described above serving as the input select signals. While this approach offers improved robustness, it is less dense than ROM storage. For example, for the Pendigits dataset (see SectionIV) that features 45 support vectors with 18 8-bit parameters each, the area, power, and latency of the ROM are 2.4cm^2, 1.9mW, and 15ms, respectively. The corresponding values for MUX-based storage are 5.0cm^2, 5.6mW, and 19ms, respectively. Although less hardware-efficient, the overhead of the MUX-based storage is not prohibitive, especially considering that this example represents the worst-case difference among all datasets examined. Therefore, we choose bespoke MUX-based storage for our SVMs to prioritize accuracy and robustness.

C. Single MAC Support Vector Engine

Our control unit makes decisions by evaluating one support vector at a time. Hence, we design a support vector engine to compute the corresponding output:

$$y = \begin{cases} 1 & \text{if } \sum_{i=1}^{m} w_i x_i + b \geq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where w_i and b are the support vector's weights and bias, and x_i are the input activations. Striving for area efficiency, we fold the computation of (1) over a single MAC unit.

As shown in Fig. 1, a register stores the accumulation result, whose size is minimized based on partial sum profiling during SVM training. A second small register is used to store a counter, which serves as the column index for the memory and for synchronization. In the first cycle, the bias is fetched, initializing the accumulation register. In each subsequent cycle, a weight is read from memory, multiplied by the respective input, and the product is accumulated. Once all model parameters are processed, the engine outputs a ready signal, providing the inverted sign of the accumulation.

D. Control Unit

OvO computes a support vector for each pair of classes, with the class attaining the highest score being selected as the SVM output. In [2], [15], all support vectors are computed in parallel, and a voter circuit determines the winning class. While this approach requires a voter, a sequential implementation also requires a $\log_2 \frac{n(n-1)}{2}$ bit counter and additional registers to track the winning class and its score. To minimize register use, we implement OvO as a decision-directed acyclic graph (DDAG) [19]–specifically, a binary decision tree–for our control Finite State Machine (FSM), requiring only a small register of $\log_2 \frac{n(n-1)}{2}$ bits to store the FSM state. Each DDAG node (FSM state) represents the current winning class, and is linked to a support vector that compares this class against a new class, yet to be considered. FSM selects the corresponding support vector by its row index in the memory. After $m+1$ cycles (where m is the number of input features), the support vector is computed in our engine, and the winner of this comparison is determined. The newly identified winning class becomes the current winner, prompting the DDAG to transition to the right if the new class prevails, or to the left otherwise.

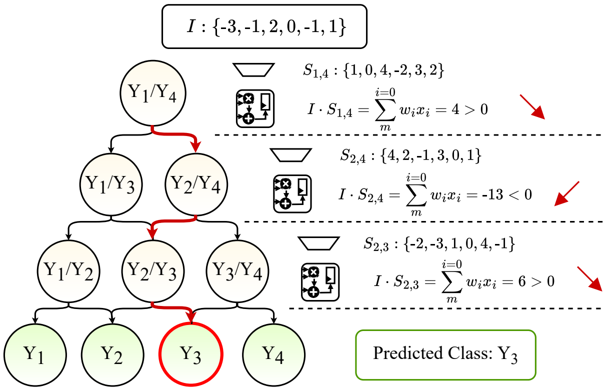
I : {-3, -1, 2, 0, -1, 1}
$S_{1,4}$: {1, 0, 4, -2, 3, 2}
$I \cdot S_{1,4} = \sum_{m}^{i=0} w_i x_i = 4 > 0$
$S_{2,4}$: {4, 2, -1, 3, 0, 1}
$I \cdot S_{2,4} = \sum_{m}^{i=0} w_i x_i = \text{-}13 < 0$
$S_{2,3}$: {-2, -3, 1, 0, 4, -1}
$I \cdot S_{2,3} = \sum_{m}^{i=0} w_i x_i = 6 > 0$
Y1/Y4
Y1/Y3
Y2/Y4
Y1/Y2
Y2/Y3
Y3/Y4
Y1
Y2
Y3
Y4
Predicted Class: Y3

Fig. 2. Example execution of our sequential SVMs. 4 classes and 6 input features/weights are considered. Support vectors corresponding to selected nodes are shown on the right, along with the classification output of the support vector engine. The output class is predicted in $3 \times 7 = 21$ cycles. Fixed point values are represented by integers.

The hardware implementation of the FSM is straightforward. It uses a hardcoded row index per state and only requires a MUX to select between two predefined next states based on the support vector engine's output. Fig. 2 presents a detailed example of our sequential execution and FSM flow. The FSM has $\frac{n(n-1)}{2}$ states (the number of OvO support vectors) but requires only $n-1$ support vector evaluations, resulting in a total of $(n-1) \times (m+1)$ cycles for each classification.

IV. Results and Analysis

In this section, we evaluate our printed sequential SVM classifiers in terms of accuracy and hardware overhead. The evaluation is based on five datasets (see Table I) from the UCI ML repository [23], selected because they involve sensor inputs relevant to printed applications [1] and are frequently used by the state of the art. We use Synopsys Design Compiler and PrimeTime for hardware evaluation, targeting frequency in the Hz range, typical for printed applications [4]. The fully parallel SVM classifiers–designed as described in [1]–serve as our baselines. Additionally, we compare against [2], [3], which, among all existing approximate printed digital ML classifiers, achieve relatively high accuracy across all datasets. To train our SVMs we use randomized parameter optimization, inputs are normalized to $[0, 1]$, and training/testing is done with a randomly distributed 80%/20% split. All accuracies hereafter

TABLE I
SVM MODEL ANALYSIS FOR THE EXAMINED DATASETS.

Dataset	FP32 Acc (%)	#Inputs	#Classes	#Support Vectors
Cardio	93.4	21	3	3
Dermatology	98.6	33	6	15
Pendigits	97.7	17	10	45
RedWine	66.2	11	6	15
WhiteWine	56.6	11	7	21

are reported on the test dataset. Table I presents the details of our trained SVMs, reporting also the FP32 software accuracy.

Table II presents the hardware evaluation of our sequential SVMs in comparison to state-of-the-art designs. As shown, our SVMs exhibit negligible to no accuracy loss compared to the FP32 model. The average area of our designs is 3.7cm^2, with a maximum of 7.8cm^2, making them realistic and suitable for most printed applications [2]. Similarly, the average power consumption is 4.2mW, with a maximum of 8.7mW, enabling all our circuits to be powered by existing printed batteries, e.g., Zinergy 15mW or BlueSpark 6mW. Table II demonstrates the feasibility and practicality of our highly accurate SVMs. Even when considering the power required for ADCs to process sensor data (e.g., 0.33mW for the required 4-bit ADC [4]), our SVMs can still operate within printed battery constraints. Since our circuits process one input feature per cycle, a single ADC can be shared across many input sensors. In contrast, ADC costs might become bottlenecks in parallel designs like [1]–[3], where inputs are processed simultaneously. We also evaluated our SVMs using a printed crossbar ROM for storage, which would reduce the area for Cardio, Dermatology, Pendigits, RedWine, and WhiteWine by 10%, 23%, 28%, 8%, and 3%, respectively. However, to fully exploit these (limited) gains, a variation-aware training approach, e.g., similar to [22], would be necessary, which does not consistently yield reliable results across all datasets.

Compared to fully parallel exact SVMs [1], as shown in Table II, our sequential SVMs achieve 10x lower area and 30x lower power, on average. For some datasets, our SVMs achieve higher accuracy due to our use of the more flexible LinearSVC class (compared to the SVC used in [1]), which provides better scalability and a wider range of penalty and loss functions. Against the approximate SVMs [2], our circuits exhibit, on average, 6x and 12x lower area and power, respectively, while attaining 4.8% higher accuracy. Compared to approximate MLPs [3], our SVMs yield 2x less area and 6x less power on average, with 6% higher accuracy. For RedWine, [3] demonstrates a lower area, but both circuits are small-enough, whereas our design achieves 10% higher accuracy. Compared to more approximate digital printed neural networks [18], [16] (not included in Table II), our SVMs feature higher area but achieve, on average, 7.4% and 7.9% higher accuracy than [18] and [16], respectively. Interestingly, for the Pendigits dataset (the most complex examined), our SVMs offer 6x lower area, 5x lower power, and 4% higher accuracy than [18], and 3x lower area and power, and 8% higher accuracy compared to [16]. Overall, as dataset complexity increases, the advantages of our sequential approach become even more pronounced, with higher gains over the state-of-the-art. Furthermore, it's important to note that, except for RedWine and WhiteWine [3], all state-of-the-art classifiers in Table II consume over 30mW, which does not align with printed batteries [1], and would be impractical for printed applications. Concluding, within the strict physical constraints of printed applications in area and power availability, our SVMs constitute the most accurate printed classifiers that successfully meet these requirements.

TABLE II
HARDWARE EVALUATION AND COMPARISON WITH STATE OF THE ART.

Dataset	Technique	Accuracy	Area	Power	Freq.
		(%)	(cm^2)	(mW)	(Hz)
Cardio	Parl SVM [1]	90.0	15.1	57.4	13
Cardio	Ax Parl SVM [2]	89.0	17.0	48.9	13
Cardio	Ax Parl MLP [3]	87.0	6.1	20.8	5
Cardio	*Ours Seq. SVM*	93.1	3.1	3.6	24
Dermatology	Parl SVM [1]	97.2	60.4	182.9	8
Dermatology	*Ours Seq. SVM*	98.6	4.9	5.3	18
Pendigits	Parl SVM [1]	97.8	123.8	364.4	4
Pendigits	Ax Parl SVM [2]	97.0	97.0	183.7	4
Pendigits	Ax Parl MLP [3]	93.0	32.7	99.2	4
Pendigits	*Ours Seq. SVM*	97.6	7.8	8.7	15
RedWine	Parl SVM [1]	57.0	23.5	92.8	15
RedWine	Ax Parl SVM [2]	56.0	11.7	21.3	15
RedWine	Ax Parl MLP [3]	56.0	1.1	3.9	5
RedWine	*Ours Seq. SVM*	66.2	3.1	3.5	19
WhiteWine	Parl SVM [1]	53.0	28.3	112.4	17
WhiteWine	Ax Parl SVM [2]	52.0	11.0	34.7	17
WhiteWine	Ax Parl MLP [3]	53.0	6.5	21.3	5
WhiteWine	*Ours Seq. SVM*	56.2	3.4	3.9	20

As shown in Table II, folding the SVM execution allows for higher clock frequencies. However, this frequency gain does not fully offset the increased latency and energy demands due to the additional cycles required. Nevertheless, this is unlikely to be an issue for most applications, as performance is not the primary focus in low-throughput printed applications [14]. Similarly, printed batteries are customizable in capacity, shape, and voltage [24], so energy concerns might be secondary compared to power availability.

V. CONCLUSION

In this paper, we present the first printed sequential SVM classifiers optimized for the strict area and power constraints of printed electronics (PE). Using our proposed sequential architecture with bespoke storage and control, and a single-MAC engine, we achieve on average 10x lower area and 30x lower power compared to fully-parallel exact SVMs, at similar accuracy. Compared to approximate printed classifiers, our approach reduces area and power by 4x and 9x, respectively, on average, while delivering a 5.4% accuracy gain. These advantages become more pronounced as dataset complexity increases. Overall, our SVM circuits represent the most accurate digital printed classifiers that meet PE's strict area and power constraints. Although we focused on EGFET technology, our design principles are applicable to similar technologies.

ACKNOWLEDGMENT

This work is 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] M. H. Mubarik *et al.*, “Printed machine learning classifiers,” in *Annu. Int. Symp. Microarchitecture (MICRO)*, 2020, pp. 73–87.

[2] 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.

[3] 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.

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

[5] 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

[6] 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

[7] R. K. Mishra *et al.*, “Simultaneous detection of salivary δ9-tetrahydrocannabinol and alcohol using a wearable electrochemical ring sensor,” *Talanta*, vol. 211, p. 120757, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0039914020300485

[8] 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

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

[10] 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

[11] 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.

[12] 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

[13] Z. Cui, *Printed electronics: materials, technologies and applications*. John Wiley & Sons, 2016.

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

[15] 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.

[16] 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.

[17] 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.

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

[19] J. Platt, N. Cristianini, and J. Shawe-Taylor, “Large margin dags for multiclass classification,” *Advances in neural information processing systems*, vol. 12, 1999.

[20] C. Marques *et al.*, “Progress Report on “From Printed Electrolyte-Gated Metal-Oxide Devices to Circuits”,” *Advanced Materials*, vol. 31, 2019.

[21] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.

[22] H. Zhao *et al.*, “Highly-bespoke robust printed neuromorphic circuits,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023.

[23] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[24] S. Lanceros-Méndez and C. M. Costa, *Printed Batteries: Materials, Technologies and Applications*. Wiley, 2018.