A Single-MOSFET Analog High Resolution-Targeted (SMART) Multiplier for Machine Learning Classification

Farid Kenarangi, *Graduate Student Member, IEEE*, and Inna Partin-Vaisband, *Senior Member, IEEE*

***Abstract*— Mixed-signal machine-learning classification has recently been demonstrated as an efficient alternative for classification with power expensive digital circuits. In this paper, a single-MOSFET analog multiplier is proposed for classifying high-dimensional input data into multi-class output space with less power and higher accuracy than state-of-the-art mixed-signal linear classifiers. A high-resolution (*i.e.*, multi-bit) multiplication is facilitated within a single-MOSFET by feeding the features and feature weights into, respectively, the body and gate inputs. High-resolution classifier that considers the decisions of the individual predictors is designed at 180 nm technology node and operates at 100 MHz in near/subthreshold region. To evaluate the performance of the classifier, a reduced MNIST dataset is generated by downsampling the MNIST digit images from 784 features to 48 features. The system is simulated across a wide range of PVT variations, exhibiting average accuracy of 92% (2% improvement over state-of-the-art), energy consumption of 67.3 pJ per classification (over 8 times lower than state-of-the-art classifiers), area of 27,570 μm^2 per binary classifier, and a stable response under PVT variations. Finally, to provide ground for future work on ultra-low-power deep and convolutional networks, scalability and robustness of the proposed multiplier is evaluated with a convolutional neural network on CIFAR-10. Similar classification accuracy with digital and SMART hardware has been observed. All the code and simulation files are available at an online public GitHub repository, https://github.com/faridken/SMART-Multiplier-for-ML.**

***Index Terms*— Machine learning hardware, mixed-signal classifiers, high resolution, high-dimensional data, multi-class classification, linear classifiers, subthreshold, ensemble learning.**

I. Introduction

ENABLING on-device near-sensor machine learning (ML) compute on power limited devices has the potential to shift paradigms in smart applications. Relevant devices include but not limited to difficult-to-reach sensors powered by energy harvesting; smart and battery-powered devices that execute always-on applications; unplugged devices that can run on batteries for years. While such devices cannot afford the computational power needed for executing advanced ML architectures, power efficient execution of linear ML algorithms on such devices is highly desired. Yet another interesting application of those simpler, yet highly efficient ML architectures, is understanding biological behavior and particularly the brain. The goal is to gain considerable insight into the neural structures through in-circuit modeling of biological systems [1]–[3]. A primary limitation of this research is the power consumption of ultra-large-scale ICs. Inventing extremely compact and power efficient, ML-dense ICs is therefore a primary goal. Finally, tinyML (a fast-growing field of study at the intersection of ML and Embedded Systems [4]) explores the tremendous potential of ultra-low-power ML hardware to unlock an entirely new class of smart applications. Many of the most recent works in this field are looking into non-NN architectures and simpler datasets due to their low compute and memory requirements [5].

Existing on-chip classifiers can be categorized into two major domains: digital and mixed-signal [6]. A digital classifier is typically fed with binary inputs (*i.e.*, features) and uses binary feature weights, all obtained by sampling and quantizing corresponding analog signals. The classification accuracy with digital classifiers increases with the increasing number of bits assigned for features and weights. These highly accurate digital classifiers however exhibit significant power consumption and physical size and are often not suitable for power limited applications, such as battery powered sensors and those other edge devices that are wirelessly powered and powered from harvested energy. Alternatively, mixed signal classifiers aim to reduce the area and power consumption of the conventional digital classifiers by directly using the analog input data for classification [7]. The inherent need for data conversion with power hungry analog-to-digital converters (ADCs) is therefore mitigated with mixed-signal classifiers [7].

Recent state-of-the-art mixed-signal classifiers typically exhibit accuracy of 90%-99% and the overall energy consumption in the range of hundreds of picojoules to hundreds of nanojoules per decision for typical image recognition datasets [7]–[13]. Emerging device technologies are also being considered for providing power and area efficient alternatives for the conventional CMOS based classifiers [14]. Accuracy of 90% and energy of 25 pJ per decision has been recently reported in [15], [16].

To enable high-resolution (*i.e.*, multi-bit) feature-weight multiplication, a theoretical framework that comprises circuits, models, heterogeneous design framework, and linearization

Manuscript received March 18, 2021; revised September 27, 2021; accepted October 27, 2021. Date of publication November 2, 2021; date of current version December 13, 2021. This article was recommended by Guest Editor T. Mohsenin. *(Corresponding author: Farid Kenarangi.)*

The authors are with the Department of Electrical and Computer Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: fkenar2@uic.edu; vaisband@uic.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JETCAS.2021.3124940.

Digital Object Identifier 10.1109/JETCAS.2021.3124940

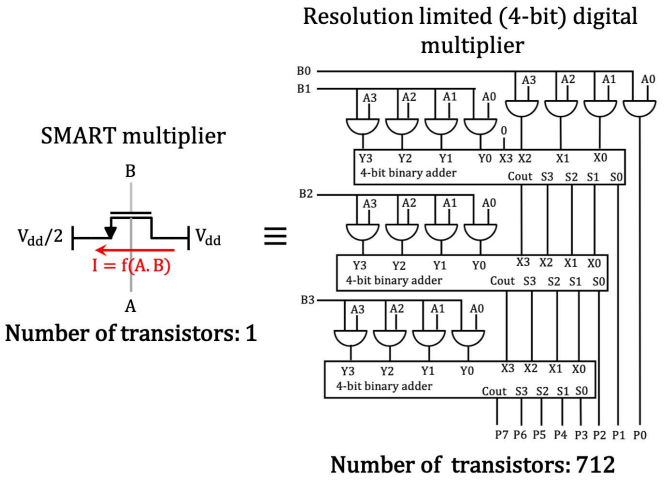

Fig. 1. Schematic of the proposed single MOSFET multiplier. Features and feature weights are fed into, respectively, the body and gate input of the individual MOS transistors and a current corresponding to the feature-weight multiplication is generated. The 4-bit digital multiplier is shown for comparison.

flow is proposed. To the best of the authors knowledge, this paper is the first to report a mixed-signal high-resolution classifier, utilizing MOSFET body terminals. The schematic of the proposed SMART multiplier is shown in Fig. 1. A typical digital multiplier is also shown in the figure for comparison. With this approach, body bias of the MOSFETs is controlled by the individual ML features, the gate inputs are fed by the absolute value of a corresponding feature weights, and the sign of the weight is considered with separate lines for the positive and negative feature and feature weight product. M-class classification with N features is therefore realized with an N-row K-column multiplication and accumulation (MAC) array, where each column serves as an independent binary classifier. These individual binary classifiers are combined using one-versus-one technique [17], requiring $K = M(M-1)/2$ binary classifiers in total. While body bias has previously been used to enhance IC characteristics (*i.e.*, in memories, amplifiers, and signal converters [2], [18]–[22]), exploiting body bias for enhancing on-chip multiplication and machine learning is novel. A primary advantage of the single-MOSFET multiplier configuration is the low sensing line capacitance, yielding low power consumption and high multiplication rate.

Another primary contribution is the heterogeneous design framework for ensemble learning. With this framework, the learning algorithms and ML hyperparameters are individually adjusted for each binary classifier based on an automated close-loop SPICE-Python feedback, enhancing the overall classification accuracy and resilience to PVT variations. With the heterogeneous approach, the individual binary classifiers are trained with unique algorithms and hyperparameters, addressing the inherent heterogeneity of typical multi-dimensional data. The preferred algorithms and corresponding hyperparameters are determined iteratively based on the feedback from SPICE simulation shell. Note that the heterogeneous training is a generic approach which can be utilized with any dataset. For a different dataset, maximum classification margin is achieved with a different set of training algorithms and hyperparatemers. The dataset-specific weights are stored within the memory and no additional hardware and/or hardware modification is required to enable the ensemble learning (see Section IV).

Finally, the proposed system is designed in near/subthreshold region, exhibiting a power efficient alternative for the traditional classifiers. The classifier is demonstrated at circuit level in SPICE with the Modified National Institute of Standards and Technology (MNIST) dataset [23] of 10-class digit images. Based on the simulation results, MNIST data is classified with 92% accuracy and 67.3 pJ per decision.

The rest of the paper is organized as follows. In Section II, the proposed high-resolution *binary* classifier and linearization technique are described. Fabrication considerations are also discussed in this section. Based on the proposed binary classifier, a *multi-class* high-resolution classifier is designed and demonstrated with MNIST dataset, as described in Section III. The proposed heterogeneous framework for enhancing classification accuracy of the multi-class classifier is described in Section IV. Circuit design and simulation results of the multi-class SMART classifier using one-versus-one technique are presented in Section V. The scalability and robustness of the SMART multiplier is evaluated in Section VI with a convolutional neural network (CNN) on CIFAR-10 dataset as a ground for future work. The paper is summarized in Section VII.

Circuit design and simulation results of the multi-class SMART classifier using one-versus-one technique are presented in Section V. The paper is summarized in Section VII.

II. The Proposed Linear Binary Classifier

In this section, the proposed linear binary classifier is described. The software level design framework is explained in Section II-A. The circuit, linearization flow, and fabrication costs are presented in, respectively, Sections II-B, II-C, and II-D.

A. Design Framework

Reliability, power consumption, and physical size of on-chip classifiers are all primary concerns in modern ML ICs. The proposed framework is designed to meet accuracy specifications of modern classification problems in a cost effective manner. Linear algorithms are exploited in this paper for training a supervised binary classifier, optimizing the system for linearly separable input data.

With a multivariate linear classifier, the system response Z is a linear combination of N input features $x = (x_1, x_2, \ldots, x_N)$ and model weights $w = (w_1, w_2, \ldots, w_N)$,

$$Z = \sum_{i=1}^{N} w_i \cdot x_i, \quad Z \in \mathbb{R}. \tag{1}$$

The model weights are determined during supervised training by minimizing the prediction error between the system response, Z, and a corresponding true value in the labeled training dataset. A combination of common supervised linear

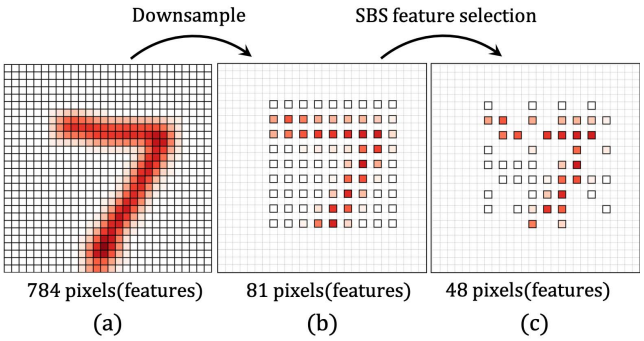

Fig. 2. Handwritten digits from MNIST dataset, (a) original image (default resolution with 28 × 28 pixels = 784 features), (b) downsampled image (9 × 9 pixels = 81 features), and (c) 48 features determined using SBS feature selection algorithm.

ML models (*i.e.*, logistic regression, passive-aggressive regression, perception, and linear support vector machine) is used to evaluate the performance of the heterogeneous SMART classifier. In inference, a probability threshold of 0.5 is used for predicting system response to input data, exhibiting a simple on-chip implementation,

$$\hat{y} = \text{sign}(Z) = \text{sign}(\sum_{i=1}^{N} w_i \cdot x_i) = \begin{cases} 1, & Z \geq 0 \\ -1, & Z < 0. \end{cases} \quad (2)$$

The accuracy of the classifier is evaluated as a percentage of all the correct predictions out of the total number of test predictions. The proposed ML flow and the preprocessing steps are explained below.

1) Dataset: MNIST database is a large image dataset, commonly used for evaluating the effectiveness of ML hardware. MNIST contains images of 70,000 handwritten digits, ranging between 0 to 9. Each digit comprises 784 (28 × 28) image pixels. The training and test datasets comprise, respectively, 60,000 and 10,000 digits. Out of the 60,000 training observations, 45,000 and 15,000 digits are used for, respectively, training and validating the proposed system.

2) Feature Selection and Downsampling: Each image pixel of the individual digits in the training set is considered as an ML feature and used for training the classifier. To reduce the power and area overheads, those redundant features that are not essential for digit classification are eliminated. A typical feature selection flow (see [7], [8]) is utilized for a fair overhead comparison. In both [7] and [8], the raw images have been low-pass filtered and downsampled from 784 to 81 features. In addition, the number of features in [7] has been further reduced to 48 using Fisher's criterion [24]. Higher number of features (*i.e.*, 81) has been used in [8] to compensate for the accuracy loss due to 1-bit resolution feature weights. In SMART classifier, the images are also filtered and downsampled to 81 features. The number of features is further reduced to 48 with the sequential backward selection (SBS) algorithm [25]. Finally, a various number of meaningful features is selected from the pool of 48 features for the individual binary classifiers within the heterogeneous framework. The original, downsampled, and 48-feature images of digit 7 are exemplified in Fig. 2.

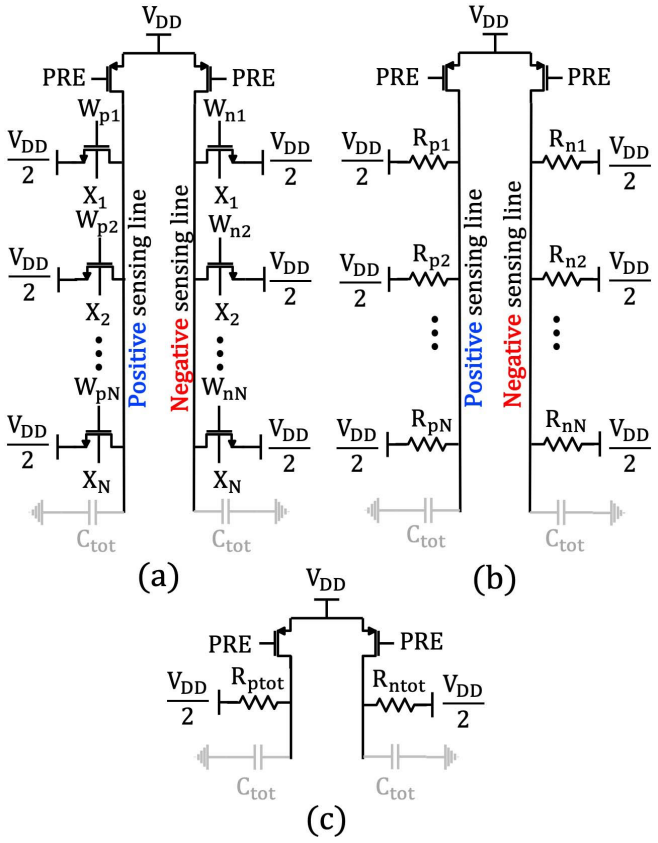

Fig. 3. SMART binary classifier, (a) the transistor level circuit diagram, (b) equivalent RC model, and (c) simplified RC model.

B. Circuit Level Considerations

The primary goal in a linear binary classification is to accurately and efficiently perform the dot product operation of the features and feature weights, as described in (2). To simplify the circuit level design, the system response is reformulated as the signed addition of positive, Z^+, and negative, Z^-, feature-weight products,

$$\hat{y} = \text{sign}(\overbrace{\sum_{w_i>0} x_i \cdot w_i}^{Z^+} + \overbrace{\sum_{w_j<0} x_j \cdot w_j}^{Z^-}) = \begin{cases} 1, & \mid Z^+ \mid \geq \mid Z^- \mid \\ -1, & \mid Z^+ \mid < \mid Z^- \mid . \end{cases} \quad (3)$$

The individual positive and negative feature-weight products are accumulated within the positive, V_{sen}^{+}, and negative, V_{sen}^{-}, sensing lines, yielding the basic ML MAC operation, as shown in Fig. 3(a). To enable a single feature-weight multiplication, a MOSFET is connected to each of the positive (*i.e.*, M_{pi}) and negative (*i.e.*, M_{ni}) sensing lines. The body terminals of the transistors in the i^{th} row (M_{pi}, M_{ni}) are connected to the corresponding feature (*i.e.*, X_i). One of the two transistors is deactivated through the grounded gate terminal. The gate terminal of the other transistor is connected to the feature weight (*i.e.*, W_i), as determined by the

feature weight sign,

$$(W_{pi}, W_{ni}) = \begin{cases} (|W_i|, 0), & W_i > 0 \\ (0, |W_i|), & W_i < 0. \end{cases} \quad (4)$$

Note that during each classification period only one MOSFET within each row is active. The other MOSFET is added to enable dynamic weight update in reconfigurable classifiers.

Intuitively, transistors with $X_i = 0$ or $W_i = 0$ should not induce a voltage drop on the sensing lines. To mitigate the current flowing through those transistors with $W_i \neq 0$ and $X_i = 0$, the source terminals of the individual MAC FETs are shorted to $V_{DD}/2$, resulting in a negative bulk-to-source voltage, $V_{BS} = -V_{DD}/2$, for $X_i = 0$. As a result, the threshold voltage is increased and the transistor is closed (for example, $V_{th}(V_{BS} = -0.9V) = V_{th}(V_{BS} = 0V) + 0.9\,\mathrm{V} = 0.48\,\mathrm{V} + 0.9\,\mathrm{V} \geq V_{GS}$ for selected range of the V_{GS} values in 180 nm technology node with power supply of 1.8 V). With this configuration, no current flows through a MAC FET with either W_i or X_i is grounded.

The classifier has two operating modes: precharge and classification. During precharge (*i.e.*, PRE = 0), the sensing lines are precharge to V_{DD}. Alternatively, during the classification mode (*i.e.*, PRE = 1), the lines are discharged by the transistors connected to the line, exhibiting a voltage drop corresponding to the magnitude of the feature-weight dot product.

The effective impedance of the MAC transistors and sensing lines plays a significant role in classification accuracy. To better understand the relation between the impedance parameters, consider an RC model of the circuit as shown in Fig. 3(b). Each transistor is modeled by its equivalent resistance (*i.e.*, R_i), as determined during the classification period. Note that the resistance of each transistor is controlled by the gate and body biases ($R_i = f(V_{GSi}, V_{BSi})$). The total resistance as seen from a sensing line is therefore,

$$\begin{aligned} R_{tot} &= R_1 || R_2 \ldots || R_N \\ &= f(V_{GS1}, V_{BS1}) || f(V_{GS2}, V_{BS2}) || \ldots || f(V_{GSN}, V_{BSN}) \\ &= f(W_1, X_1) || f(W_2, X_2) || \ldots || f(W_N, X_N), \end{aligned} \quad (5)$$

where N is the total number of features. Alternatively, the total capacitance of a line is dominated by the interconnect capacitance, C_{line}, and the MOSFET gate capacitance, C_{G0},

$$C_{tot} = N \times (C_{line0} + C_{G0}), \quad (6)$$

where C_{line0} is the capacitance of the line per feature. The model is simplified, as shown in Fig. 3(c), exhibiting for each sensing line the voltage drop,

$$V_{drop}(t) = V_{DD}\left(1 - \exp\left(\frac{-t}{R_{tot}C_{tot}}\right)\right). \quad (7)$$

where t is the time since the start of the classification mode and the time constant $R_{tot}C_{tot}$ is the discharge rate, as determined by (5) and (6). Note that while the width, W, of the individual MAC transistors only weakly affects the V_{drop} in (7) (*i.e.*, $R_{tot}C_{tot} \approx W/W = 1$), the biases of the individual transistors primarily affect the classifier resistance, but not the capacitance. To accurately classify data with an intrinsic line capacitance, maximum permissible load (*i.e.*, R_{tot}, C_{tot}) is determined based on (5)-(7). Thus, low or high values of R_{tot} (as determined by a combination of features and weights) can negatively impact the classifier performance. For example, if more than a single line fully discharges prior to vote extraction, the prediction becomes random.

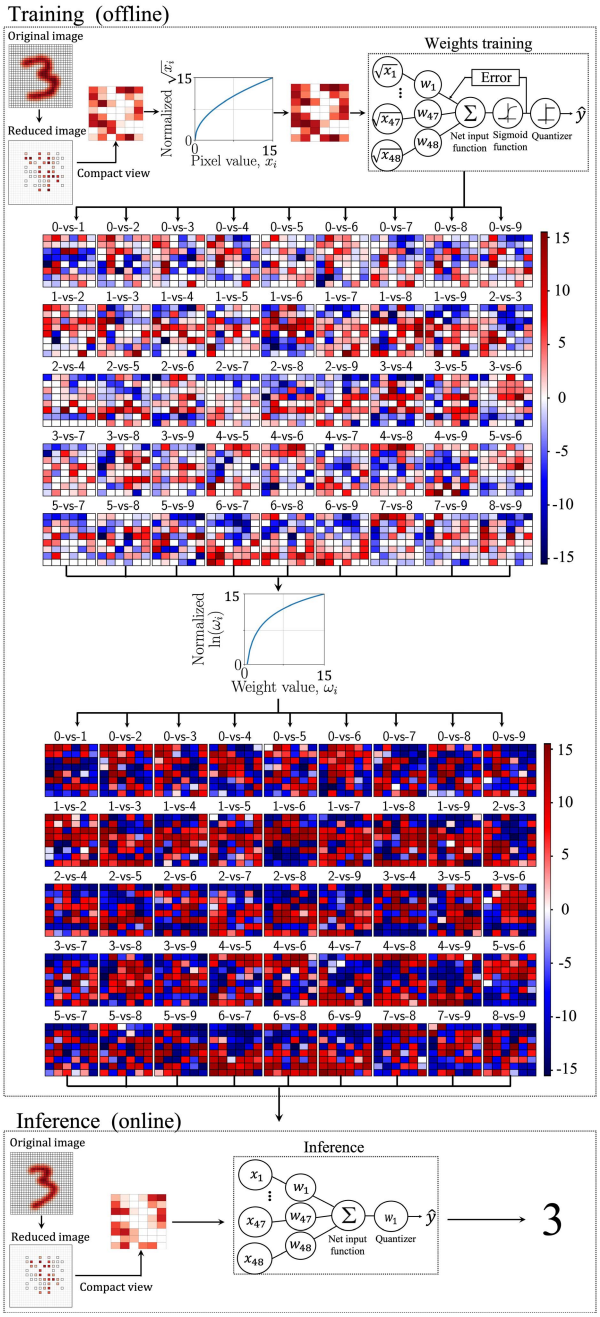

Fig. 4. Linearization flow. To account for the non-linear dependence of the drain current on the body bias, $I_{sub} \propto \sqrt{V_{bs}}$, the model is trained to make predictions based on the square root values of the original features. The optimized weights are logarithmically adjusted, mitigating the exponential dependence of the near/subthreshold current on the gate bias.

C. Linearization Flow

To provide a power efficient and scalable solution, the transistors are biased in near/subthreshold operation region,

significantly limiting the current through the sensing lines. A primary concern with near/subthreshold operation is the exponential dependence of the drain current on the body and gate biases [26],

$$I_{sub} = \frac{W}{L} I_t \exp(\frac{V_{gs} - V_{th}}{nV_T})[1 - \exp(-\frac{V_{DS}}{V_T})], \quad (8)$$

where I_t is the sub-threshold current at $V_{gs} = V_{th}$, n is the sub-threshold slope, and V_T is the thermal voltage. Note that body voltage dependence is embedded in the threshold voltage, $V_{th} \propto \sqrt{V_{bs}}$,

$$V_{th} = V_{th0} + \left(\sqrt{2\phi_f - V_{bs}} - \sqrt{2\phi_f}\right) \quad (9)$$

where V_{th0} is the threshold voltage when $V_{bs} = 0\,\text{V}$ and $2\phi_f$ is the surface potential of the silicon. Please note that the drain current in weak inversion region is practically independent of the V_{DS} voltage for $V_{DS} \gg V_T$ as the last term in (8) approaches unity [26], thus, the expression in (8) can be simplified as,

$$I_{sub} = \frac{W}{L} I_t \exp(\frac{V_{gs} - V_{th}}{nV_T}), \quad V_{DS} \gg V_T. \quad (10)$$

To mitigate the non-linear dependence of the drain current on the weight-feature dot product (see (1)), a novel training flow is proposed (see Fig. 4). To account for the square-root dependence of the drain current on the bias voltage, the model is trained with square root values of the default features ($x_i \rightarrow \sqrt{x_i}$). Thus, the extracted feature weights, w, are optimized for classifying the MNIST dataset transformed into half-order polynomial space. Alternatively, to account for the exponential dependence of the drain current on the gate voltage, the feature weights are logarithmically adjusted ($w_i \rightarrow \ln(w_i)$), yielding $V_{gs} \propto \ln(w)$. Based on the first-order approximation of $\exp(\sqrt{V_{bs}}) \approx 1 + \sqrt{V_{bs}}$, the current in this case is expressed as,

$$I_{sub} \propto \exp\left(\ln\left(V_{gs}\right)\right) \exp\left(\sqrt{V_{bs}}\right) \propto V_{gs}\sqrt{V_{bs}} \propto w\sqrt{x}. \quad (11)$$

In inference, the current model is exploited for making prediction based on the square root values of the original features, as trained offline, yielding 92% accuracy across the MNIST test set, as detailed in the following sections.

D. Fabrication Costs

In the proposed linear binary classifier, the body and gate terminals are fed by, respectively, the input features and corresponding feature weights. Each multiplication is, therefore, executed by a single-MOSFET, significantly reducing the power and area costs (despite the overhead of the triple-well technology) and complexity (as determined by number of transistors) of the classifier in comparison to the existing state-of-the-art mixed-signal classifiers [7]–[13].

Conventional twin-well fabrication process is illustrated in Fig. 5(a). This process is designed to provide a single voltage connection to all the n-type and p-type body terminals. Alternatively, to independently control the body terminals of the individual multipliers, a specialized fabrication process is required. One way to independently bias numerous body terminals, is with triple-well fabrication process (see Fig. 5(b)), which is commonly used in high-performance, low-power ICs [27], [28] and for reducing substrate noise in mixed-signal circuits [29]. With a p-substrate triple-well process, an additional deep n-well is diffused to isolate the p-well diffusions of the individual MOSFETs from the common p-substrate, allowing an independent body terminal connection for each MOSFET. The triple-well structure yields better noise characteristics as compared with the traditional twin-well structure, without increasing the gate leakage [30]. Alternatively, the triple-well structure, exhibits additional fabrication costs and area overheads due to requirements on the minimum width and spacing of deep n-wells. The minimum permissible width and spacing of deep n-wells for 180 nm PDK is, respectively, 3 μm and 5 μm [31], resulting in five times larger area than with the twin-well process. In this paper, the triple-well overheads are determined based on transistor layout and considered as part of the reported results.

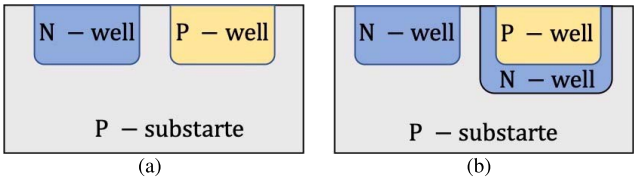

Fig. 5. Common fabrication processes with CMOS technology and p-substrate, (a) twin-well process, and (b) triple-well process with p-substrate.

III. Smart Multi-Class Classifier

A multi-class classifier is designed based on the linear binary classifiers, as presented in Section II. One-versus-one (OVO) approach is preferred to address the integrity of multi-class classification, as described in Section III.A. The transistor level implementation of the proposed SMART classifier is presented in Section III.B.

A. OVO Multi-Class Classification Scheme

Two typical approaches for designing a multi-class classifier based on multiple binary classifiers are one-versus-one (OVO) and one-versus-all (OVA) [32]. With OVA approach, each binary classifier discriminates between a single class and the rest of the classes. The required number of binary classifiers with OVA increases linearly with the number of classes. Alternatively, with OVO approach, all pairwise combinations of the output classes are evaluated with the individual binary classifiers. An M-class classification with OVO approach requires $M(M-1)/2$ binary classifiers. For classifying MNIST dataset ($M = 10$) with OVO approach, 45 i-verus-j, ($i, j \in \{0, 1, 2, .., 9\}$) binary classifiers are therefore required. The final decision with OVO technique is extracted using majority voting approach [33]: each binary classifier votes independently for a certain class and the final decision is made based on the class with highest number of votes.

While OVA is a more power and area efficient classification scheme as fewer number of binary classifier needs to be utilized (*i.e.*, $(M-1)/2$ times less), it typically exhibits lower accuracy and poor performance under PVT variations as compared to the OVO scheme. In this paper, OVO scheme

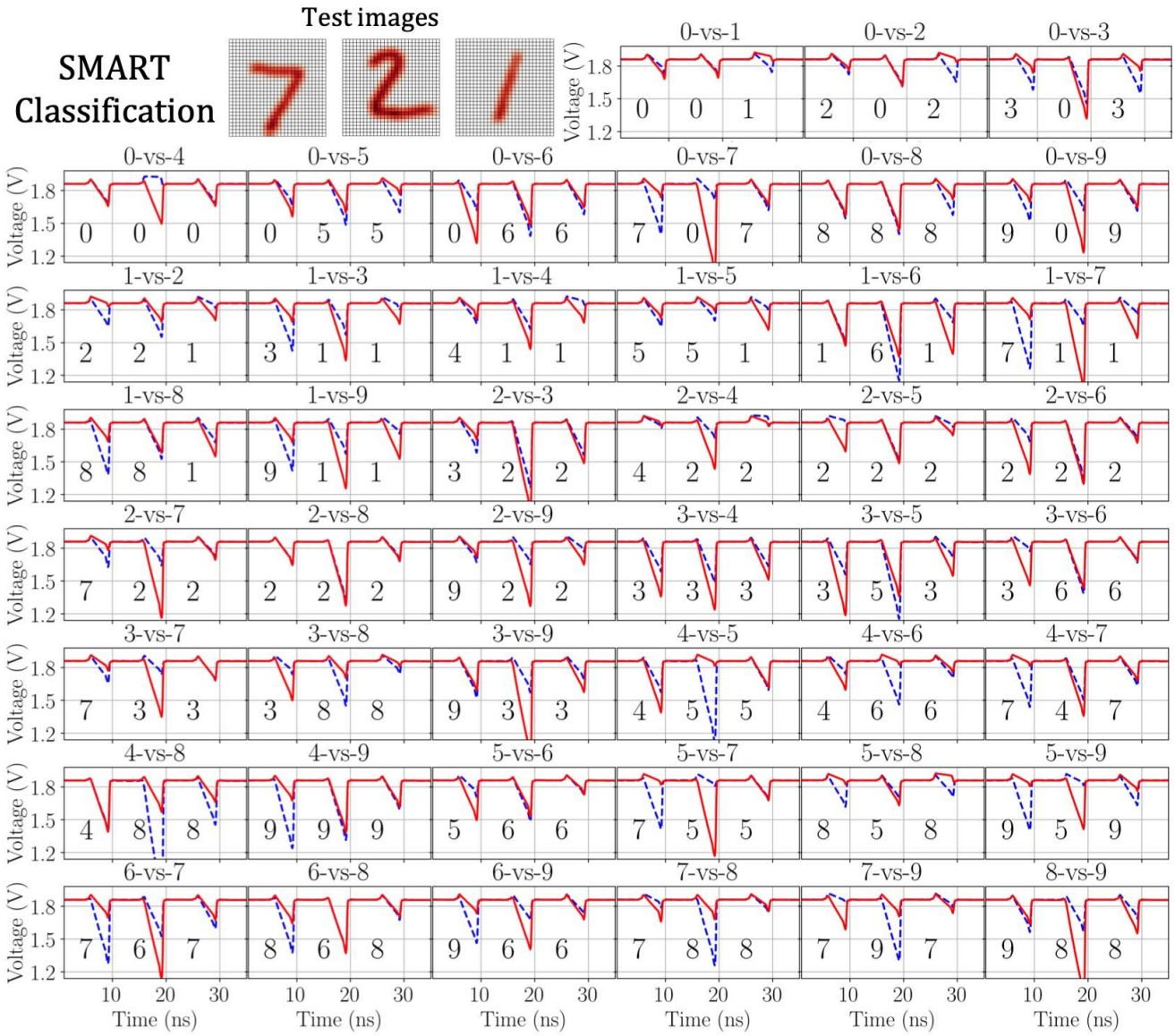

Fig. 6. Voltage waveforms of the sensing lines during the precharge (*i.e.*, (0, 6), (10, 16), and (20, 26) nanosecond time intervals) and classification stages (*i.e.*, (6, 10), (16, 20), and (26, 30) nanosecond time intervals) for three consecutive input features (*i.e.*, 7, 2, and 1). The votes extracted during each period are shown below the waveforms.

is preferred over OVA to maximize the classification accuracy. The vote for each binary classifier i-verus-j (*i.e.*, $vote(i, j)$) is determined based on the relative difference of the voltage drops across the corresponding positive (*i.e.*, $\Delta V^{+}_{sen}(i, j)$) and negative (*i.e.*, $\Delta V^{-}_{sen}(i, j)$) sensing lines,

$$vote(i, j) = \begin{cases} 1, & \Delta V^{+}_{sen}(i, j) > \Delta V^{-}_{sen}(i, j) \\ 0, & \Delta V^{+}_{sen}(i, j) < \Delta V^{-}_{sen}(i, j). \end{cases} \tag{12}$$

To extract the classifier vote at the circuit level, a sensing amplifier is designed, as described in the next subsection.

B. Circuit Level Design and Simulation Results

The proposed multi-class classifier is designed in SPICE and demonstrated based on the reduced MNIST dataset. The OVO circuits and the architecture of the MOSFET array are described in this section.

1) MAC Array: To classify the downsampled MNIST digits, 45 binary classifiers (see Section II) are co-designed in SPICE, yielding a MAC array of at most 48 $\times$ 45 simultaneously activated transistors. Each of the transistors within the MAC array is exploited for generating a single feature-weight product. During inference, the V^{+}_{sen} and V^{-}_{sen} lines are precharged to V_{DD} prior to each prediction. All the input features and feature weights are connected simultaneously to, respectively, the body and gate terminals of the individual multiplier transistors, facilitating a parallel classification process within all the 45 binary classifiers. As a result, 45 $\times$ 2 different voltage drop values (*i.e.*, $\Delta V^{+}_{sen}(i, j)$, $\Delta V^{-}_{sen}(i, j)$, $i, j = 0, 1, \ldots, 9$) are generated on the individual sensing lines, as shown in Fig. 6. The voltage waveforms of the positive and negative sensing lines are illustrated by, respectively, the blue dotted and solid red lines. These voltage drops are sensed with a comparator to generate the classifier decision (*i.e.*, vote). The obtained

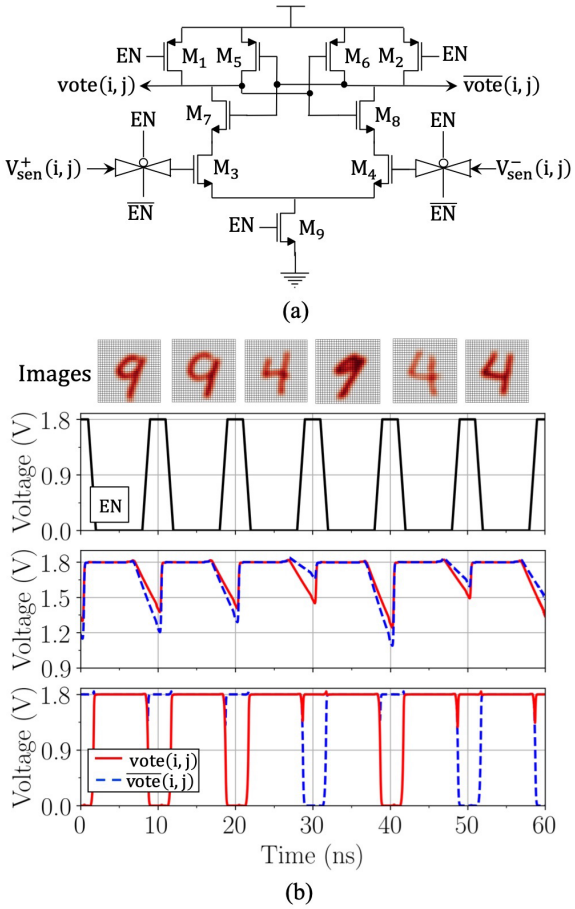

Fig. 7. (a) Schematic of the vote extractor. (b) The votes of the i-vs-j binary classifier are exemplified for $i = 4$ and $j = 9$.

votes for each classification period (as determined based on (12)) are also shown in Fig. 6 for each binary classifier. Finally the predicted digit is determined based on the digit with highest number of votes. For the examples shown in Fig. 6, the digits 7, 2, and 1 exhibit the highest number of votes (*i.e.*, 9, 9, and 9), resulting in three correct predictions.

2) Vote Extractor: To extract the vote of each binary classifier, a sensing amplifier (*i.e.*, vote extractor) is designed [34]. The schematic of the vote extractor for a multi-class classification is presented in Fig. 7(a). For a M-class classification, $M(M-1)/2$ vote extractors are required, yielding a total of 45 extractors for the MNIST dataset ($M = 10$). Each vote extractor compares the voltage levels of the positive and negative sensing lines ($V^{+}_{sen}(i, j)$ and $V^{-}_{sen}(i, j)$) and identifies the line with higher voltage drop, as shown in (12).

Initially (*i.e.*, $EN = 0$), the outputs (*i.e.*, $vote(i, j)$ and $\overline{vote}(i, j)$) are precharged to V_{DD} by the pull-up network (*i.e.*, M_1 and M_2). During the voting stage (*i.e.*, $EN = 1$), the pull-down network (*i.e.*, M_3 and M_4) is activated. Depending on the relative strength of the signals at the gate terminals of the pull-down network (*i.e.*, $V^{+}_{sen}(i, j)$ and $V^{-}_{sen}(i, j)$), M_3 sinks higher or lower current than M_4. Finally, the current difference is sensed by the back-to-back inverter (*i.e.*, M_5, M_6, M_7, and M_8) and the vote is generated. For example, if the voltage on the positive sensing line is higher than the voltage on the negative sensing line, the left branch (*i.e.*, M_3) will sink higher current, forcing the left (*i.e.*, $vote(i, j)$) and right (*i.e.*, $\overline{vote}(i, j)$) branches to, respectively, 0 and V_{DD}. Voltage waveforms of the sensing lines, EN signal which enables voter extractor, and output signals are illustrated in Fig. 7(b) for six consecutive classification periods.

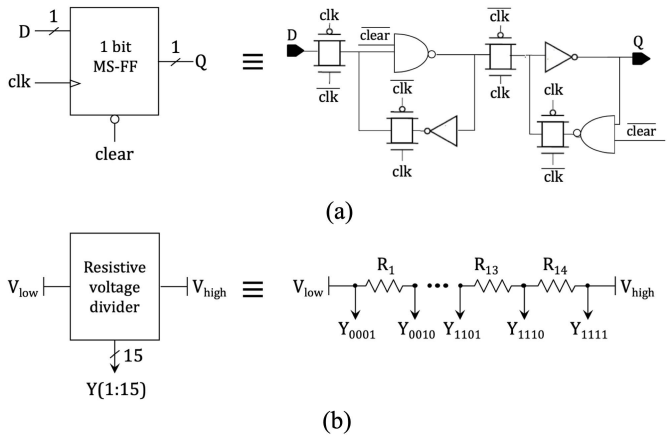

Fig. 8. Schematic of the (a) memory unit, and (b) the resistive voltage divider.

3) Resistive Voltage Divider: The quantized features and trained feature weights are stored in Master-Slave Flip-Flops (MS-FF), as shown in Fig. 8(a). To generate the quantized voltage levels for features and feature weights, resistive voltage dividers are utilized as shown in Fig. 8(b) [35]. Poly resistors ($R = 100\Omega$) with sheet resistance of 7.5 $\Omega/\square$ are utilized within the resistive voltage dividers. In this configuration, the preferred voltage range [V_{low}, V_{high}] is divided into $2^n - 1$ equal steps, where n is the preferred quantization resolution. Multiplexers are designed between the resistive voltage divider and MS-FF memory to enable system reconfigurability (see Fig. 9(a)).

To reduce the power consumption of the classifier and the overall load on the sensing lines, the MAC array is biased in near/subthreshold region. To satisfy this condition, the maximum gate-source voltage, V_{GS}, is limited by the threshold voltage (*i.e.*, $V_{th} = 0.48\,\text{V}$) for all the MAC array transistors. To quantize the feature weights with a step size of 20 mV, 4-bit resolution is considered, limiting the minimum V_{GS} to $V_{th} - (2^4 - 1) \times 20\,\text{mV} = 0.18\,\text{V}$ (*i.e.*, $0.18\,\text{V} \leq V_{GS} \leq 0.48\,\text{V}$). Note that the source terminals are shorted to $V_{DD}/2$ (*i.e.*, $V_S = 0.9\,\text{V}$), limiting the gate voltage to $1.08\,\text{V} = 0.18\,\text{V} + 0.9\,\text{V} \leq VG \leq 0.48\,\text{V} + 0.9\,\text{V} = 1.38\,\text{V}$, as shown in Fig. 9(b). Similarly, the 4-bit resolution is considered to quantize the features with a step size of 40 mV, limiting the body bias to $0.9\,\text{V} \leq V_B \leq 0.9\,\text{V} + (2^4 - 1) \times 40\,\text{mV} = 1.5\,\text{V}$, as shown in Fig. 9(c).

IV. The Proposed Heterogeneous Classification Framework

The power and area efficient, mixed-signal classifiers typically exhibit high sensitivity to circuit nonidealities and variation sources that can negatively impact the performance. To enhance the classification performance, a hardware/software co-optimization framework is proposed.

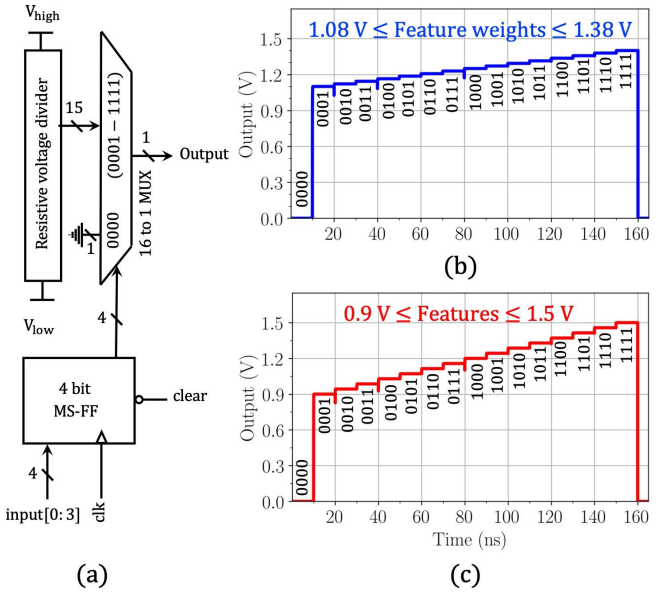

Fig. 9. Signal quantization with DAC, (a) schematic of the circuit, comprising a resistive voltage divider, multiplexer, and memory, (b) quantized feature weights, as observed at a DAC output, and (c) feature voltage levels, as observed at another DAC output. The 4-bit resolution yields 40 mV and 20 mV voltage steps for, respectively, the features and feature weights with the selected signal ranges.

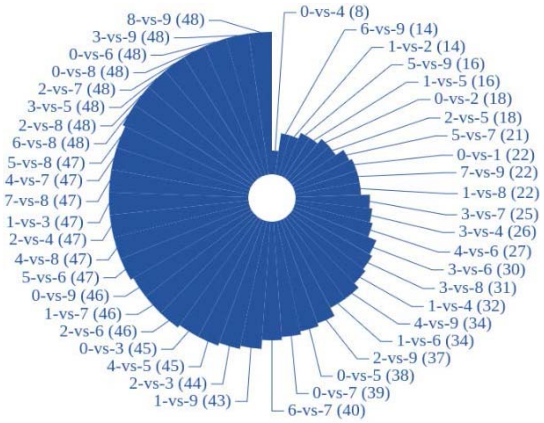

Fig. 10. Number of selected features for each binary classifier (*i.e.*, i-vs-j), increasing in the clockwise direction. Visually similar digits (*e.g.*, 8-vs-9) require more features than easily distinguishable digits (*e.g.*, 0-vs-4).

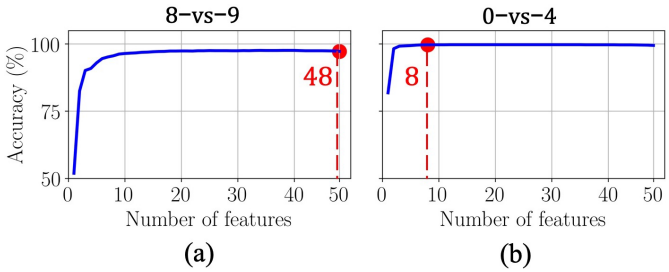

Fig. 11. Accuracy curves and minimum number of features. The preferred number of features (yielding the 48-feature accuracy) is shown for, (a) high complexity (8-vs-9), and (b) low complexity (0-vs-4) classification.

The framework exploits ensemble learning to increase the classification noise margins, minimizing the system sensitivity to variations. The unique optimization of the number of features, ML algorithm, and hyperparameters for each binary classifier is further motivated by the independence of the votes of the individual binary classifiers with the OVO scheme. The optimization process is described in the following subsections.

A. Optimizing the Number of Features

The pairwise digit classification is performed with 45 binary classifiers. The complexity of the classification task varies from one binary classifier to another. For example, the complexity of the 8-vs-9 classification is high due to visual similarity of "8" and "9," as determined by the low data variance between these digits. Alternatively, the digits "0" and "4" exhibit higher data variance and are easier to discriminate, yielding a lower classification complexity. The number of features required for an accurate binary classification therefore increases with the higher complexity of the classification task. The number of preferred features (up to 48, as determined with the SBS algorithm), is shown in Fig. 10 for each binary classifier. For a binary classifier i-vs-j, this number is determined as the *minimum* number of features required to discriminate between the digits i and j without accuracy degradation (as compared to classification with 48 features). The rate of accuracy degradation with smaller number of features varies among the binary classifiers, yielding a different minimum for each classifier, as exemplified in Fig. 11.

B. Optimize the ML Parameters

The SMART binary classifiers are designed to perform feature-weight dot product (see (1)). Any ML algorithm that is based on (1) can be exploited within a SMART classifier to train the weights. Examples of SMART compatible ML algorithms are logistic regression (LR), passive-aggressive regression (PAR), perception (PER), and linear support vector machine (LSVM) [36]. With mixed-signal multi-class classifiers, all binary classifiers are typically trained with the same ML algorithm. Alternatively, with the proposed design framework, a preferred learning algorithm is individually determined for each classifier, maximizing the classification margin and noise resilience. A pool of learning algorithms (*i.e.*, logistic regression, passive-aggressive regression, perception, and linear support vector machine) and hyperparameters (*i.e.*, regularization penalty (α), regularization rate ($\lambda = 1/C$), and loss function (L1, L2, and Hinge) [36]) is considered. The preferred training algorithms and hyperparameters are determined for each binary classifier based on the proposed iterative Python-SPICE simulation framework, as listed in Fig. 12.

Note that the effect of hyperparameters on classifier performance is different for circuit level simulation with quantized feature weights and Python simulation on a conventional digital platform with floating point (FP) weights. In particular, loss function and regularization rate guide the distribution of the feature weights and consequently, the accuracy degradation due to feature weight quantization [37]. With digital accelerators, classifier's generalization capacity increases with the increasing regularization up to the point of underfitting the data. Thus, a preferred regularization rate exists, yielding max-

Logistic regression | Passive-aggressive regression | Perception | Linear support vector machine

i-vs-j	1	2	3	4	5	6	7	8	9
0	L2, C=10^{-3}	Hinge, C=10^{-3}	L2, C=10^{-5}	L2, C=10^{-4}	L2, C=10^{-5}	L2, α=10^{-5}	L2, C=10^{-4}	Hinge, C=10^{-5}	L1, C=10^{-2}
1		L1, C=10^{-2}	L1, C=10^{-1}	L2, C=10^{-7}	L2, α=10^{-5}	L2, C=10^{-6}	L1, C=10^{-3}	L2, α=10^{-7}	L1, C=10^{-3}
2			L2, C=10^{-7}	L1, C=10^{-4}	L2, C=10^{-5}	L2, α=10^{-5}	L2, C=10^{-3}	L2, C=10^{-7}	L1, C=10^{-3}
3				L2, α=10^{-9}	L2, C=10^{-6}	L1, C=10^{-3}	Hinge, C=10^{-7}	L2, C=10^{-3}	L2, C=10^{-6}
4					L1, C=10^{-2}	L1, C=10^{-4}	L1, C=10^{-3}	L2, C=10^{-6}	L2, C=10^{-6}
5						L2, C=10^{-1}	L1, C=10^{-3}	L1, C=10^{-4}	L1, C=10^{-4}
6							L2, C=10^{-1}	L2, C=10^{-3}	L2, C=10^{-6}
7								L2, C=10^{-2}	Hinge, C=10^{-5}
8									L2, C=10^{-5}

Fig. 12. ML algorithms and hyperparameters as determined within the heterogeneous design framework. The cells are shaded based on the individually selected ML algorithms. The first value in each cell is the loss function (L1, L2, or Hinge). The second value is the regularization parameter (α or $C = 1/\lambda$).

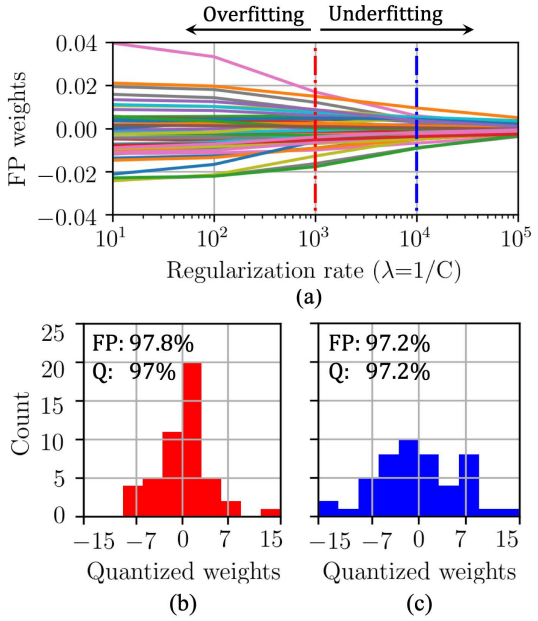

Fig. 13. Accuracy tradeoffs between regularization and quantization. (a) Feature weights extracted for the 5-vs-6 classifier with different values of regularization rate ($\lambda = 1/C$) in floating point (FP) format. (b) Lower regularization yields higher classification accuracy with FP weights (97.8% > 97.2%). (c) Higher regularization yields higher classification accuracy with quantized weights (97.2% > 97%).

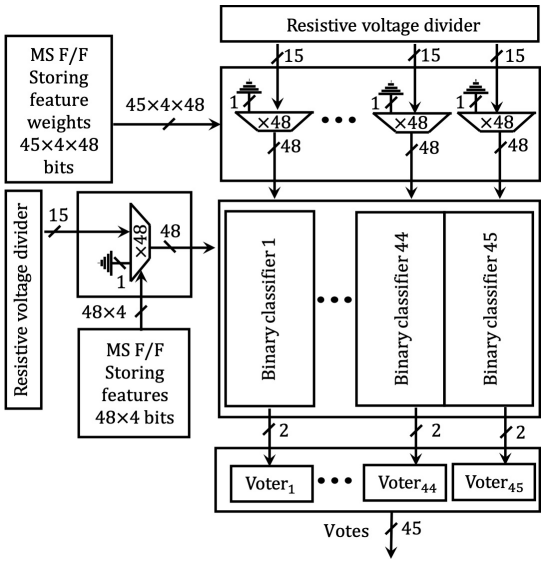

Fig. 14. Schematic of the proposed classifier, comprising memory, voltage divider, multiplexer, MOSFET array, and vote extractor.

imum accuracy with FP weights. However, when quantized, majority of similar weights are mapped into the same values, exhibiting suboptimal utilization of the available weight range. Alternatively, moderately underfitted FP models (with low-variance weights at higher regularization rates) yield a higher number of distinguishable weights when quantized over a fixed weight range. Thus, the optimum regularization rate tends to be higher with quantized weights than the optimum rate with FP weights.

To illustrate the quantization-regularization tradeoff, consider the 5-vs-6 binary classifier trained with logistic regression and various regularization rates, $\lambda = 1/C$. FP and 4-bit quantized weights and the respective accuracies in Python and at the circuit level are shown in Fig. 13. While the training with $\lambda = 10^3$ yields highest accuracy in Python (see Fig. 13(b)), higher regularization ($\lambda = 10^4$) is preferred for maximizing the accuracy at the circuit level (see Fig. 13(c)).

V. Results

A. System Characteristics

A schematic of the integrated system is illustrated in Fig. 14, comprising memory, resistive voltage divider, multiplexers, MAC array, and vote extractors. The proposed SMART classifier is designed using the 180 nm PDK in Cadence Virtuoso with the nominal power supply voltage of 1.8 V and threshold voltage of 0.48 V. The body and gate terminals of the MAC array are biased at voltage levels of, respectively, [0.9 V, 1.5 V] and [1.08 V, 1.38 V]. The area occupied per binary classifier is 27,570 μm^2, as estimated based on transistor count in SPICE. The MAC array comprises a total of $48 \times (45 \times 2) = 4,320$ MOSFETs. The reduced MNIST dataset, as shown in Fig. 10 is classified with 92% accuracy within a single 10 ns clock cycle of the system operation, exhibiting no accuracy degradation as compared with the validation accuracy in Python. The confusion matrices obtained based on Python and SPICE simulations are shown in, respectively, Fig. 15(a) and Fig. 15(b), exhibiting equal accuracy of 92%. No performance degradation due to quantization and linearization is therefore observed. The ML classifier generates predictions at 100 MHz frequency, exhibiting an average energy consumption of 67.3 pJ per

TABLE I

SYSTEM CHARACTERISTICS OF THE PROPOSED AND OTHER EXISTING MIXED-SIGNAL ML CLASSIFIERS

		TCAS-I 2017 [7]	JSSC 2017 [8]	TCAS-II 2020 [9]	ASSCC 2016 [10]	JETCAS 2020 [11]	TCAS-I 2020 [12]	JETCAS 2018 [13]	This work
Type of the results		Fabrication	Fabrication	Simulation	Fabrication	Simulation	Simulation	Simulation	Simulation
Technology (nm)		130	130	40	28	65	65	65	180
Dataset		MNIST	MNIST	MNIST	MNIST	MNIST	MNIST	MNIST	MNIST
Algorithm		Ada-boost linear	Ada-boost linear	DNN non-linear	ANN non-linear	LetNet-5 non-linear	LetNet-5 non-linear	LetNet-5 non-linear	Ensemble of linear classifiers
Accuracy (%)		90	90	98	98	97	99	97	92
Feature weight resolution		4	1	6	8	2	5	8	4
Feature resolution		5	5	8	8	5	5	6	4
Supply voltage (V)		1.2	1.2	1.1	1.0	1.2	1.2	1.0	1.8
Speed (decisions/sec)		1.3M	50M	25M	2.4M	2.5M	0.34M	0.07M	100M
Costs per classification	Energy (pJ)	543	633	38,000	N/A	302,280	158,203	450,000	67.3
	Energy per MAC (fJ)	51	6	380	200	360	254	900	42
	Electric charge (pA·sec)*	425	528	34,000	N/A	251,900	131,836	450,000	37
	Electric charge per MAC (fA·sec)	45	5	340	200	300	212	900	22

* Electric charge = Energy / Supply voltage

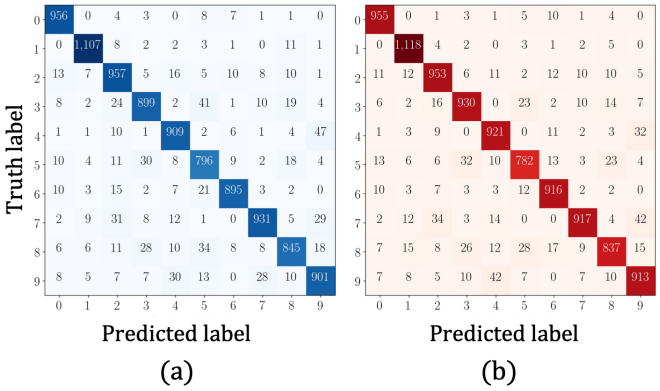

Fig. 15. Confusion matrix of the MNIST classification obtained in (a) Python (92% accuracy), and (b) SPICE (92% accuracy), exhibiting no accuracy degradation in SPICE as compared with Python results.

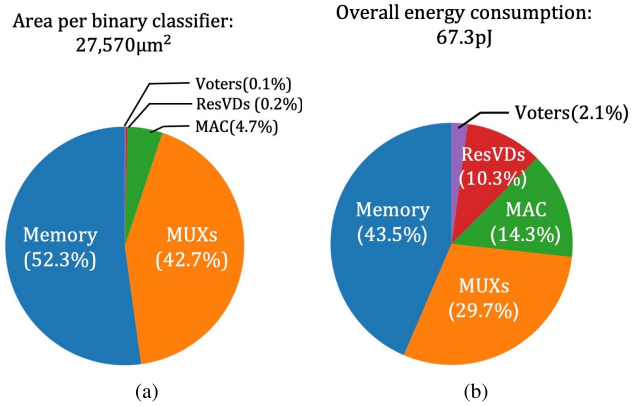

Fig. 16. The overhead with the SMART classifier, (a) physical area is dominated by memory and MUXs, and (b) power is primarily dissipated by the memory, MUXs, MAC array, and resistive voltage dividers (ResVDs).

classification of a single digit. To maintain high prediction accuracy, four bits are assigned for quantizing, the feature weights and input features. By increasing the dimensionality of the proposed classifier, lower power and area overheads can be traded off for higher prediction accuracy, approaching the theoretical limit of 94% for MNIST classification with linear ML algorithms and OVO decisioning scheme. The power and area overhead breakdown of the SMART classifier is illustrated in Fig. 16.

B. Simulation Results

Performance characteristics are listed in Table I for the proposed system along with the existing state-of-the-art mixed-signal classifiers [7]–[13]. To the best of the authors' knowledge, MNIST classification accuracy with SMART classifier is higher by 2% than the classification accuracy with any reported mixed-signal linear classifier [7], [8]. Benefiting from the single MOSFET multiplication, SMART classifier exhibits significantly less MAC transistors and sensing line capacitance, resulting in over an order of magnitude lower power consumption, as compared with the state-of-the-art classifiers. The operational frequency is scalable and can be adjusted based on application needs and constrains. For fair comparison, the overall electric charge (current × time) per decision is also shown in Table I. The electric charge per decision is eleven times lower with SMART classifier as compared with other approaches. Owing to the significantly lower transistor count and otherwise typical size of the auxiliary circuits, the proposed classifier is expected to exhibit favorable size characteristics as compared to other similar systems. To avoid a biased comparison between the size of systems demonstrated at transistor level and in-silicon, estimated area characteristics are not included in Table I. Additionally, to avoid bias towards the selected ML algorithm, electric charge per MAC is also included in Table I. A comparison of the electric charge per MAC, speed, and accuracy for the proposed and state-of-the-art classifiers is shown in Fig. 17.

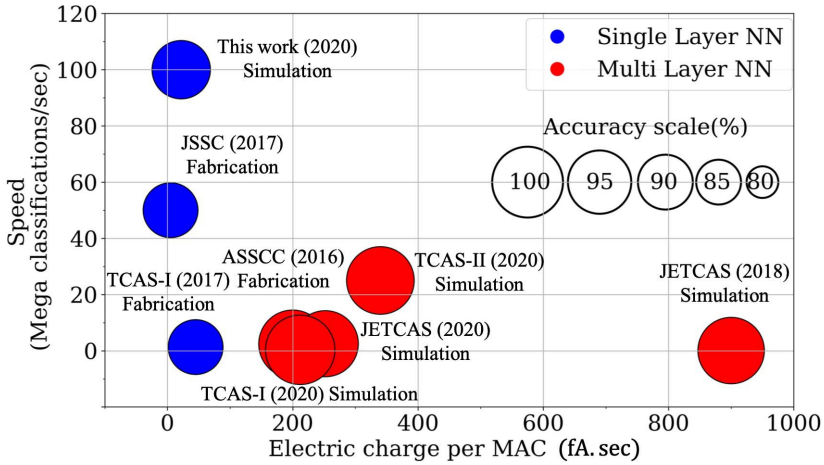

Fig. 17. Electric charge per MAC, classification speed, and accuracy of the proposed and state-of-the-art classifiers.

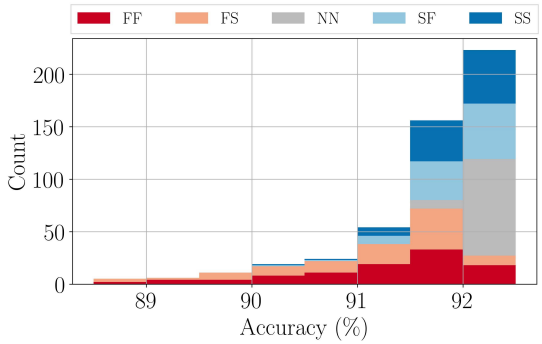

Fig. 18. Classifier performance under PVT and corner variations ($1.62\,\text{V} < V_{DD} < 1.98\,\text{V}, -30\,^{\circ}\text{C} < T < 125\,^{\circ}\text{C}$).

To evaluate the effect of process, voltage, temperature (PVT) and corner variations with the SMART classifier, the supply voltage is varied between 1.62 volts and 1.98 volts (*i.e.*, 10% variations) and the temperature is varied between $-30\,^{\circ}\text{C}$ and $125\,^{\circ}\text{C}$ across all the operating corners (*i.e.*, FF, FS, NN, SF, and SS). The effect of the combined PVT, mismatch, and corner variations is evaluated with a 500-run Monte-Carlo simulation on a randomly selected 1,000-observation (100 images per digit) balanced test set with nominal accuracy of 92%. Note that a 500-run Monte-Carlo simulation on the whole test set takes 500×0.5 hours on Intel Core i7-7700 CPU. Based on the simulation results (see Fig. 18), the classifier exhibits no sensitivity to PVT variations at nominal corner (*i.e.*, average accuracy is 92%). Furthermore, less than 1% accuracy degradation is observed at FF, FS, SF, SS corners. The accuracy degradation at fast and slow corners is dominated by the MAC array. At FF corner, the MAC transistors discharge the sensing lines at a faster (than nominal) rate. In those FF cases when both the positive and negative lines are depleted prior to vote extraction, an erroneous result is recorded. Alternatively at SS corner, the MAC transistors discharge the sensing lines at a slower (than nominal) rate, exhibiting no sensible voltage difference at vote time and thus, an erroneous classification result. To increase the accuracy at FF and SS corners, real-time detection of the operational corner and adaptive frequency scaling should be considered (*i.e.*, the classifier should be operated at higher and lower frequencies at, respectively, FF and SS corners). Finally, note that due to the comparative nature of the vote extraction scheme, high resilience to noise is expected, similar to other comparison-based classifiers [38].

VI. Future Work–SMART Multiplier for Advanced NN Architectures

While linear classification with the proposed SMART multiplier is the primary objective of this work, evaluating the SMART multiplier with various ML architectures and datasets is of interest for future work. Particularly, the effectiveness of SMART-based CNN in classification of CIFAR-10 dataset is evaluated in this section. Please note that the objective in this section is to provide ground for future work on ultra-low-power deep and convolutional networks, rather than to exhaustively demonstrate classification of CIFAR-10. To evaluate the robustness and scalability of the proposed framework, the SMART multiplier is analytically modeled, and a CNN is developed based on the model and tested on CIFAR-10 dataset.

Each SMART matrix multiplier comprises a single sensing line and multiplier transistors connected to the sensing line. To capture the technology-specific transistor behavior, a single transistor is simulated in SPICE with various gate and body biases and the resistance is extracted and interpolated. Sensing line capacitance is assumed to be fixed, and the multiplier transistors are connected in parallel. Thus, the total resistance, capacitance, and the overall voltage drop on the sensing line can be analytically determined based on the sensing line voltage in (7) at any point of time, yielding the analytical SMART multiplier model used to simulate SMART CNN in this section. This model has been exhaustively simulated for MNIST classification, yielding less than 2% deviation as compare with SPICE simulation results.

Two operations often required by ML algorithms are convolution and fully connected (FC) operations. While a single fully connected layer is utilized to classify the MNIST dataset in this work, convolution can be conveniently implemented using unrolling (*i.e.*, expansion using multiple matrix multiplications). Similarly, the proposed SMART multiplier can

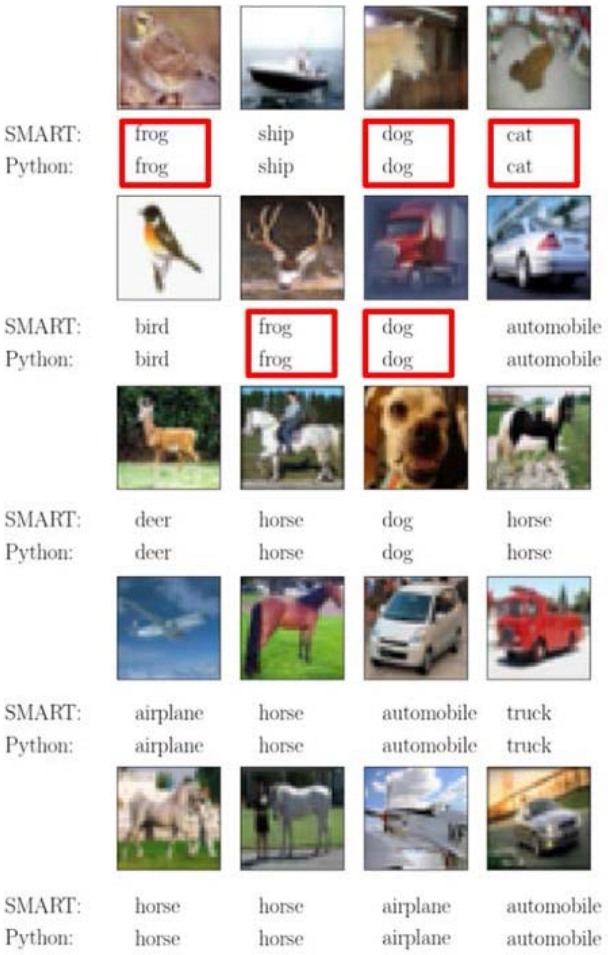

Fig. 19. Prediction results for 20 consecutive CIFAR-10 images. The true label images and the SMART and Python classifications are shown. Albeit the 25% error (due to shallow CNN), no difference between classifications with digital and SMART hardware has been observed.

be extended to perform convolution and be utilized in more complex algorithms. To design a convolution layer using the SMART multiplier, convolution operation is unrolled using interdigitated loops. Note that while the SMART convolution exhibits 10%-15% deviation from the ideal convolution results, this does not have a critical effect on the overall CNN accuracy, as shown in this section. This is in line with the recent analytical results that show that multiplication-based operations can be approximated while additions need to be exact in CNN MAC operations [39].

To show the ability of the SMART multiplier to classify CIFAR-10 data with accuracy of a digital accelerator, a four-layer CNN is designed with two convolutional, two FC layers, and a total of over 800,000 trainable parameters. Note that to demonstrate digital-like (rather than the highest possible) accuracy of SMART classification, a shallower CNN architecture is preferred for the sake of shorter runtime of computationally intensive IC simulation. Typical layers such as batch normalization, max pooling, dropout, and flatten layers are simulated in Python. The classifier is trained using TensorFlow in Python, exhibiting 75% accuracy on a digital accelerator. No feature selection or preprocessing has been utilized while preparing the data. The prediction results from Python and with the SMART multiplier are shown in Fig. 19 for 20 randomly selected CIFAR-10 images. Note that the classification accuracy with this CIFAR-10 subset is also 75%. In this configuration, the SMART classifier exhibits the same predictions as Python, demonstrating the robustness and scalability of the SMART multiplier in designing complex DNN and CNN networks.

VII. Conclusion

Several state-of-the-art mixed-signal classifiers have recently been demonstrated for power efficient classification. Accurate classification of multi-dimensional data under the tight power and area constraints is the primary objective in modern on-chip classifiers. A novel circuit topology is proposed in this paper for ML classification based on a single-MOSFET analog, high resolution-targeted (SMART) multiplier. With this topology, the body terminal of the each MAC MOSFET is exploited to encode input features, enabling the high-resolution classification. To the best of the authors knowledge, the proposed SMART classifier is the first integrated system to successfully classify MNIST dataset in near/subthreshold region using a single-MOSFET MAC. Biasing transistors in near/subthreshold region significantly decreases the leakage and dynamic currents as well as the overall load on the sensing lines. OVO decisioning scheme is exploited to accurately extract the final decision based on the votes of multiple binary classifiers. A heterogeneous design framework is developed to determine the learning algorithms and ML hyperparameters for the individual binary classifiers, increasing the accuracy and resilience to PVT variations.

The proposed SMART classifier is designed in SPICE and simulated in 180 nm standard CMOS process. The performance and functionality is validated with simulation results, exhibiting 92% (2% higher than the state-of-the-art) classification accuracy with 67.3 pJ energy consumption per prediction with MNIST dataset. Each prediction is finalized within a single clock cycle of 10 ns. The unique topology of SMART classifier supports the ML integrity under a wide range of PVT, mismatch, and corner variations, as well as system scalability across technology nodes. It is also expected to enable a power-efficient ML compute in more complex deep and convolutional networks. All the code and simulation files are available at an online public GitHub repository [40].

References

[1] G. Indiveri, B. Linares-Barranco, and M. Payvand, "System-level integration in neuromorphic co-processors," in *Memristive Devices for Brain-Inspired Computing*. Oxford, U.K.: Woodhead, 2020, pp. 479–497.

[2] A. Kumar *et al.*, "A 0.065-mm^2 19.8-mW single-channel calibration-free 12-b 600-MS/s ADC in 28-nm UTBB FD-SOI using FBB," *IEEE J. Solid-State Circuits*, vol. 52, no. 7, pp. 1927–1939, Jul. 2017.

[3] *Blue Brain Project*. Accessed: Nov. 4, 2021. [Online]. Available: https://www.epfl.ch/research/domains/bluebrain/

[4] *TinyML*. Accessed: Nov. 4, 2021. [Online]. Available: https://www.tinyml.org/

[5] C. R. Banbury *et al.*, "Benchmarking TinyML systems: Challenges and direction," 2020, *arXiv:2003.04821*.

[6] V. Sze, "Designing hardware for machine learning: The important role played by circuit designers," *IEEE Solid State Circuits Mag.*, vol. 9, no. 4, pp. 46–54, Nov. 2017.
[7] Z. Wang and N. Verma, "A low-energy machine-learning classifier based on clocked comparators for direct inference on analog sensors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 11, pp. 2954–2965, Nov. 2017.
[8] J. Zhang, Z. Wang, and N. Verma, "In-memory computation of a machine-learning classifier in a standard 6T SRAM array," *IEEE J. Solid-State Circuits*, vol. 52, no. 4, pp. 915–924, Apr. 2017.
[9] H. Xiong, M. A. Bakar, and G. He, "Hardware implementation of an improved stochastic computing based deep neural network using short sequence length," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 11, pp. 2667–2671, Nov. 2020.
[10] D. Bankman and B. Murmann, "An 8-bit, 16 input, 3.2 pJ/op switched-capacitor dot product circuit in 28-nm FDSOI CMOS," in *Proc. IEEE Asian Conf. Solid-State Circuits*, Nov. 2016, pp. 21–24.
[11] A. Agrawal, A. Kosta, S. Kodge, D. E. Kim, and K. Roy, "CASH-RAM: Enabling in-memory computations for edge inference using charge accumulation and sharing in standard 8T-SRAM arrays," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 10, no. 3, pp. 295–305, Sep. 2020.
[12] M. Ali, A. Jaiswal, S. Kodge, A. Agrawal, I. Chakraborty, and K. Roy, "IMAC: In-memory multi-bit multiplication and accumulation in 6T SRAM array," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 8, pp. 2521–2531, Aug. 2020.
[13] M. Kang, S. Lim, S. Gonugondla, and N. R. Shanbhag, "An in-memory VLSI architecture for convolutional neural networks," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 3, pp. 494–505, Sep. 2018.
[14] F. Kenarangi, X. Hu, Y. Liu, J. A. C. Incorvia, J. S. Friedman, and I. Partin-Vaisband, "Exploiting dual-gate ambipolar CNFETs for scalable machine learning classification," *Sci. Rep.*, vol. 10, no. 1, pp. 1–10, Mar. 2020.
[15] D. Querlioz, O. Bichler, and C. Gamrat, "Simulation of a memristor-based spiking neural network immune to device variations," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1775–1781.
[16] F. Kenarangi and I. Partin-Vaisband, "Leveraging independent double-gate FinFET devices for machine learning classification," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4356–4367, Nov. 2019.
[17] S. Knerr, L. Personnaz, and G. Dreyfus, "Single-layer learning revisited: A stepwise procedure for building and training a neural network," in *Neurocomputing*, vol. 68. Berlin, Germany: Springer, 1990, pp. 41–50.
[18] H. Mostafa, M. Anis, and M. Elmasry, "Adaptive body bias for reducing the impacts of NBTI and process variations on 6T SRAM cells," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 12, pp. 2859–2871, Dec. 2011.
[19] D.-S. Lee, Y.-H. Jun, and B.-S. Kong, "Simultaneous reverse body and negative word-line biasing control scheme for leakage reduction of DRAM," *IEEE J. Solid-State Circuits*, vol. 46, no. 10, pp. 2396–2405, Oct. 2011.
[20] M. Sumita S. Sakiyama, M. Kinoshita, Y. Araki, Y. Ikeda, and K. Fukuoka, "Mixed body bias techniques with fixed V_t and I_{ds} generation circuits," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 60–66, Jan. 2005.
[21] B. Liu, J. Cai, J. Yuan, and Y. Hei, "A low-voltage SRAM sense amplifier with offset cancelling using digitized multiple body biasing," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 4, pp. 442–446, Apr. 2017.
[22] A. Biswas *et al.*, "Energy-efficient smart embedded memory design for IoT and AI," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, Jun. 2018.
[23] Y. LeCun. (1998). *The MNIST Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[24] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.
[25] S. Raschka, *Python Machine Learning*, 1st ed. Birmingham, U.K.: Packt Publishing, 2015.
[26] P. R. Gray, P. Hurst, R. G. Meyer, and S. Lewis, *Analysis and Design of Analog Integrated Circuits*, 4th ed. New York, NY, USA: Wiley, 2001.
[27] J. W. Tschanz *et al.*, "Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage," *IEEE J. Solid-State Circuits*, vol. 37, no. 11, pp. 1396–1402, Nov. 2002.
[28] R. Taco, I. Levi, M. Lanuzza, and A. Fish, "Low voltage logic circuits exploiting gate level dynamic body biasing in 28 nm UTBB FD-SOI," *Solid State Electron.*, vol. 117, pp. 185–192, Mar. 2016.
[29] K. H. To *et al.*, "Comprehensive study of substrate noise isolation for mixed-signal circuits," in *IEDM Tech. Dig.*, Dec. 2001, pp. 7–22.
[30] Y. Ogasahara, M. Hashimoto, T. Kanamoto, and T. Onoye, "Supply noise suppression by triple-well structure," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 4, pp. 781–785, Apr. 2013.
[31] TSMC. *0.18-Micron (um) Technology*. Accessed: Nov. 4, 2021. [Online]. Available: https://www.tsmc.com/english/dedicatedFoundry/technology/logic
[32] M. Aly, "Survey on multiclass classification methods," *Neural Netw.*, vol. 19, pp. 1–9, Nov. 2005.
[33] J. Levin and B. Nalebuff, "An introduction to vote-counting schemes," *J. Econ. Perspect.*, vol. 9, no. 1, pp. 3–26, Feb. 1995.
[34] T. Kobayashi, K. Nogami, T. Shirotori, and Y. Fujimoto, "A current-controlled latch sense amplifier and a static power-saving input buffer for low-power architecture," *IEEE J. Solid-State Circuits*, vol. 76, no. 5, pp. 863–867, May 1993.
[35] C.-W. Lu, P.-Y. Yin, C.-M. Hsiao, M.-C. F. Chang, and Y.-S. Lin, "A 10-bit resistor-floating-resistor-string DAC (RFR-DAC) for high color-depth LCD driver ICs," *IEEE J. Solid-State Circuits*, vol. 47, no. 10, pp. 2454–2466, Oct. 2012.
[36] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.
[37] M. Wess, S. M. P. Dinakarrao, and A. Jantsch, "Weighted quantization-regularization in DNNs for weight memory minimization toward HW implementation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 11, pp. 2929–2939, Nov. 2018.
[38] T. Sepke, P. Holloway, C. G. Sodini, and H.-S. Lee, "Noise analysis for comparator-based circuits," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 3, pp. 541–553, Mar. 2009.
[39] M. S. Kim, A. A. Del Barrio Garcia, H. Kim, and N. Bagherzadeh, "The effects of approximate multiplication on convolutional neural networks," *IEEE Trans. Emerg. Topics Comput.*, early access, Jan. 12, 2021, doi: 10.1109/TETC.2021.3050989.
[40] F. Kenarangi and I. P.-Vaisband. *Source Code and Simulation Files*. Accessed: Nov. 4, 2021. [Online]. Available: https://github.com/faridken/SMART-Multiplier-for-ML

Farid Kenarangi (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Tabriz, Tabriz, Iran, in 2015. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Illinois at Chicago, Chicago, IL, USA, under the supervision of Prof. I. Partin-Vaisband. His current research interests include hardware security, machine learning integrated circuits, analog design, and on-chip power delivery and management. He was a recipient of the 2017 University of Illinois at Chicago Chancellor's Graduate Research Award.

Inna Partin-Vaisband (Senior Member, IEEE) received the B.Sc. degree in computer engineering and the M.Sc. degree in electrical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 2006 and 2009, respectively, and the Ph.D. degree in electrical engineering from the University of Rochester, Rochester, NY, USA, in 2015.

From 2003 to 2009, she held a variety of software and hardware research and development positions at Tower Semiconductor Ltd., G-Connect Ltd., and IBM Ltd., all in Israel. Her primary interest includes the area of high performance integrated circuit design. She is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of Illinois at Chicago. Her research is currently focused on innovation in the areas of AI hardware and hardware security. Yet another primary focus is on distributed power delivery and locally intelligent power management that facilitates performance scalability in heterogeneous ultra-large scale integrated systems. Her special emphasis is placed on developing robust frameworks across levels of design abstraction for complex heterogeneous integrated systems. She is an Associate Editor of the *Microelectronics Journal* and has served on the technical program and organization committees of various conferences.