IEEE Open Journal of
Circuits and Systems

Received 23 January 2024; revised 30 April 2024; accepted 28 May 2024. Date of publication 30 May 2024; date of current version 17 June 2024.

Digital Object Identifier 10.1109/OJCAS.2024.3407663

A Radar-Based System for Detection of Human Fall Utilizing Analog Hardware Architectures of Decision Tree Model

VASSILIS ALIMISIS[1] (Student Member, IEEE), DIMITRIOS G. ARNAOUTOGLOU[2] (Graduate Student Member, IEEE), EMMANOUIL ANASTASIOS SERLIS[1] (Student Member, IEEE), ARGYRO KAMPERI[1] (Student Member, IEEE), KONSTANTINOS METAXAS[1] (Student Member, IEEE), GEORGE A. KYRIACOU[2] (Senior Member, IEEE), AND PAUL P. SOTIRIADIS[1] (Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece
[2]Department of Electrical and Computer Engineering, Democritus University of Thrace, 67100 Xanthi, Greece

This article was recommended by Associate Editor M. Dandin.

CORRESPONDING AUTHOR: V. ALIMISIS (e-mail: alimisisv@mail.ntua.gr)

This work was supported in part by the Project "Study, Design, Development and Implementation of a Holistic System for Upgrading the Quality of Life and Activity of the Elderly" under Grant MIS 5047294, which is implemented under the Action "Support for Regional Excellence," funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014–2020) and co-financed by Greece and the European Union (European Regional Development Fund). The publication of the article in OA mode was financially supported by HEAL-Link.

ABSTRACT A fall-detection system was implemented utilizing a 2.45 GHz continuous wave radar along with power-efficient and fully-analog integrated classifier architectures. The Power Burst Curve and the effective acceleration were derived from the short time Fourier transform, and then processed by the analog classifier. The proposed classifier architectures are based on different approximations of the Decision tree classification model. The architectures consist of three main building blocks: sigmoid function circuit, analog multiplier and an argmax operator circuit. To assess the hardware design, a thorough analysis is performed, comparing it to commonly used analog classifiers while exploiting the extracted data. The architectures were trained using Python and were compared to software-based classifiers. The circuit designs were executed using TSMC's 90 nm CMOS process technology and the Cadence IC Suite was employed for tasks including design, schematic implementation, and post-layout simulations.

INDEX TERMS Analog hardware classifier, decision tree, fall-detection, radar-based system, sigmoid-based implementation, sub-threshold region.

I. INTRODUCTION

FALL detection systems are essential for safeguarding individuals, particularly those prone to accidental falls, like the elderly or those with mobility challenges [1], [2]. These systems utilize a range of sensors and cutting-edge technologies to swiftly identify instances of falls, enabling prompt intervention and potentially preventing severe injuries [3], [4]. The advancement of robust fall detection technology is a significant focal point in healthcare and assistive technologies, with progress in sensor technology and machine learning algorithms contributing to more precise and dependable systems [1], [2]. Ultimately, this enhances the quality of life and independence for those requiring additional support and care.

Biomedical engineering has played a pivotal role in advancing fall detection technology, especially in the realm of healthcare and assistive technologies [5]. By integrating state-of-the-art sensors, advanced signal processing techniques and machine learning algorithms, biomedical engineers have developed sophisticated systems capable of accurately detecting and responding to falls [3], [4]. These systems leverage various sensors, including accelerometers [6], [7], gyroscopes [8] and sometimes radar-based technology [9], [10], [11] to monitor

individuals' movements and orientation. Through meticulous design and implementation, these technologies can effectively differentiate between routine activities and a genuine fall event, facilitating prompt intervention [12]. The integration of biomedical engineering and fall detection not only enhances the safety of vulnerable populations but also underscores the transformative potential of technology in improving healthcare outcomes and overall quality of life.

Analog computing has emerged as a valuable tool in the field of biomedical engineering, offering unique advantages in processing and analyzing biological data [13], [14], [15]. The continuous nature of analog signals is particularly well-suited to modeling and simulating biological systems, where physiological processes often exhibit intricate and dynamic behavior [16]. Biomedical engineers have leveraged analog computing to create high-precision analog electronic circuits that can mimic and study physiological responses, such as neural activity, cardiac rhythms or biomechanical movements [17], [18]. These analog models enable researchers to gain deeper insights into complex biological phenomena, providing a bridge between mathematical modeling and real-world biological systems [13], [14], [15]. Furthermore, analog computing can be employed in specialized medical devices, including analog signal processing for diagnostic tools, patient monitoring and data acquisition, contributing to enhanced healthcare solutions and improving patient outcomes [19], [20]. The synergy between analog computing and biomedical engineering continues to drive innovation in the development of novel diagnostic and therapeutic approaches in the realm of healthcare and life sciences.

Motivated by the need for energy-efficient and compact biomedical smart sensor systems in fall detection [21], [22], [23], this study introduces a Radar-based system for detecting human falls. This system consists of a Radar-based method and a low-power analog integrated decision tree classification model. Regarding the Radar-based method is described in two simultaneously submitted articles: one herein (Data Collection) and the other [24]. The work in [24] elaborates on the technical aspects and on how to deliver a real time estimation of an elder fall. Therein, for the final classification method was achieved with software-based Support Vector Machine (SVM), thus needing an external processor unit (e.g., laptop) to carry out the computational burden thus the size, cost and power dissipation are not optimized. On the contrary, the effect herein is focused on minimizing the power consumption along with delivering a very fast processing of the collected data from the radar operation. For this purpose, the classification step is implemented into an appropriately designed fully analog hardware classifier. Thus, the effort herein is directed toward the design of this classifier with particular attention to optimizing its performance. These two articles present complementary approaches which could ultimately be combined in an advanced system.

The implemented fall-detection method was tested utilizing the processed radar's data, which are described in more details in [24], with ultimate purpose to operate in real-time scenario. Instead of using a Support vector machine [24], the extracted data (acceleration and power burst curve) are subsequently processed by the proposed low-power analog decision tree classification model. Post-layout simulations conducted in a TSMC 90 nm CMOS process via Cadence IC Suite validate the accuracy of the implemented classifiers. Moreover, the proposed classifiers are compared with a software-based implementation and related analog classifiers in the measured dataset.

The rest of the paper is organized as follows. The motivation and our vision is explained in Section II. The proposed fall-detection method based on CW Radar system is analysed in Section III. The related background concerning the Decision tree classification model is presented in Section IV. Section V refers to the proposed design methodology, which includes the system level architecture, along with the transistor level implementation of the main circuits used. The offline training and architecture's parametrization is analysed in Section VI. In Section VII, the post-layout simulation result in the measured data are provided along with the implemented layout. A comparison study between this work and related analog-based classification systems is summarized in Section VIII. Finally, Section IX concludes the article.

II. MOTIVATION AND VISION

Indoor radars offer immunity from external factors such as acoustical noise, lighting, and smoke, and enable "seeing" through different objects or walls, allowing for accurate detection and ensuring privacy in almost every situation. This stands in contrast to camera-based systems, which are sensitive to these factors [25]. A continuous wave (CW) radar system offers a simple structure and operation, making it perfect for real-life applications where target localization is not required. In [24], an acceleration-based fall detection technique was proposed utilizing a CW radar consisting of a Software Defined Radio (SDR), a mid-range processing unit (a mid-range laptop with an Intel Core i5-10210U processor), and two antennas. A simplified block diagram of the system's operation is depicted in Fig. 1. In summary, the SDR generates the signal in the digital domain, converts it to analog, and transmits it using the antenna. Next, the scattered signal from the target is collected through the antenna and processed in the analog domain before being converted back into digital form. In most cases, the digital system's processing is performed on an external processing unit (e.g., a laptop) due to the limiting capabilities of the SDR's internal processing unit. Subsequently, the Short-Time Fourier Transform (STFT) and feature estimation are applied, and the features are inputted into a digital classifier (e.g., Support Vector Machine) to determine if a fall has occurred [24].

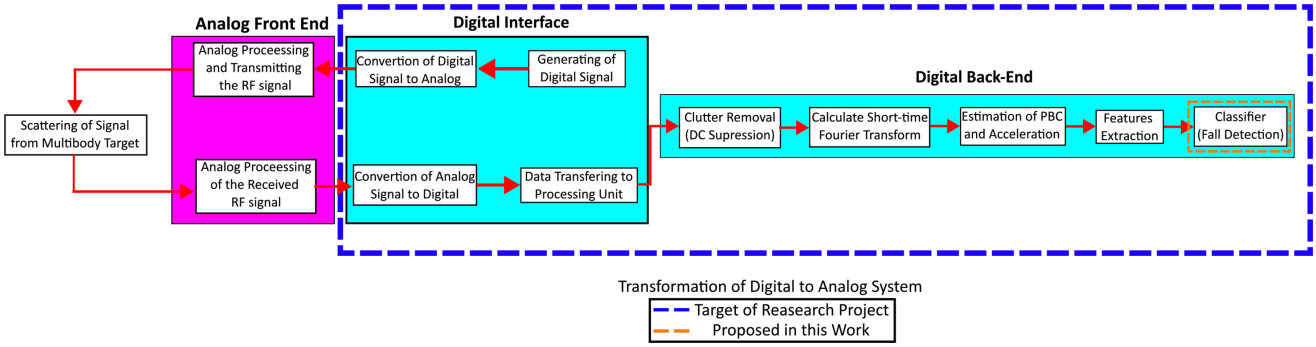


FIGURE 1. Radar Fall Detection System based on Acceleration [24] and Proposed Analog Transformation Modifications.

However, this method is power-hungry due to the multiple conversions between the analog and digital domains and the constant operation of the digital processing unit. Our vision is the development of a novel, fully-analog, power-efficient fall detection system. To achieve this goal, as depicted in Fig. 1 with the dotted line, we aim to replace the digital interface and back-end with analog IC blocks. Specifically, the SDR will be substituted with an analog radar system, eliminating the need for the digital interface and energy-consuming converters [26]. It is important to note, that analog processing will occur in the baseband or Intermediate Frequency, rather than in the RF signal, because CW radar does not require high bandwidth to operate, thus allowing for down-conversion in the analog front end [27]. In more details, the clutter removal in analog signal is well-established method from the introduction of radar systems [27]. In past years, in the literature different works were proposed about calculating Fast Fourier Transform (FFT) in analog domain [28], [29], [30]. The procedure to estimate the metrics (PBC and acceleration) includes summations, products and differentiations which can be implemented with translinear circuits [31], [32] or low-power approaches [33], [34]. The same operations will be utilized to numerically approximate feature extraction with a degradation in accuracy to reduce the power consumption of the overall system [35], [36], [37], [38]. Finally, the classifier will be a purely analog energy-efficient approach with a power consumption lower than μW.

In this study, building upon our previously published work [24] where the system methodology was validated using a CW radar based on SDR, we introduce a novel fully-analog classifier to replace the digital counterpart. Following this initial step, once the functionality of the analog classifier is confirmed, we will proceed to implement the remaining blocks of the digital back-end using analog components. In the end, the entire system will be evaluated under real-time conditions to assess its overall performance and average power consumption.

III. DATA COLLECTION

Measurements were conducted in the facilities at Department of Electrical and Computer Engineering of Democritus

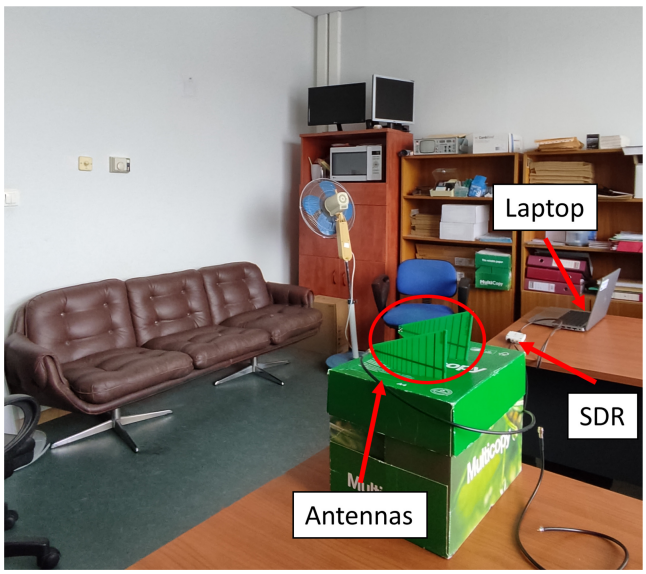


FIGURE 2. Experimental site (office) where the data set measurements were recorded by the CW radar.

University of Thrace in Xanthi. The central operating frequency was determined to belong to free industrial scientific and medical (ISM) band 2.45 GHz. An office was selected as the experimental site where the SDR (USRP B205-mini) and the laptop were positioned on a desk and the antennas were located in the next desk as depicted in Fig. 2. This specific USRP is a general low cost and footprint transceiver with limited computational power and operational bandwidth. It is connected through a USB port to a portable computer (e.g., laptop) with a power consumption around 2.5 W. The power consumption is almost constant due to the use of Variable Attenuators in RF analog front end for power control. This is a common practise in RF modules, because the design and implementation of power variable microwave sources and amplifiers is almost impossible. The data set includes a total of 187 different cases of daily activities such as walking around, sitting in a chair, kneeling and falling in different angles. We selected to perform 37 fall cases and 150 non-fall cases because the fall cases will be unlikely to happen and we do not want the algorithm to be biased. To

verify the validity of our system Line-of-Sight (LoS) tests were conducted with an approximately duration of 10 s each one by a volunteer undergraduate student. We selected the LoS conditions in order to test the accuracy of the proposed analog classifiers based on the presented metrics. In a future work, we will test their operation for LoS and Non LoS cases in the same dataset. It is important to note that in Line-of-Sight tests the low power level Wi-fi signal was not interfered with the relative higher power radar signal. The average distance of the antennas and the volunteer in each case was 2.5 m while the maximum distance was defined at 5 m. The fall cases were performed with the limitation of not being perpendicular to the propagation of the electromagnetic wave in order to be able to detect a radial velocity due to Doppler's effect.

The raw data from radar were collected and then processed, as detailed analyzed in [24], and the power burst curve (PBC) and human's effective acceleration were derived. In summary, the CW signal travels through free space, encounters a multibody target (e.g., person), and gets scattered. Some of this scattered signal is reflected back and captured by the receiving antenna. This received signal is then amplified and converted back to digital for processing. The processing involves removing the influence of stationary targets by suppressing the DC component using a moving average, from which real-time adjustments are subtracted. Additionally, a Short Time Fourier Transform (STFT) is applied to determine the Doppler-shifted frequency response indicative of moving targets over time. The system calculates a power burst curve and its moving average to determine if the signal exceeds a certain threshold, indicating movement. If so, acceleration is computed based on a specified methodology [24].

The PBC expresses the power per time instant [39] and is a good way to determine intense moving of the target (such as human fall) [40]. This is a measure of the signal energy, in the spectrum within a specific frequency band and is estimated summing at each time step the power of each frequency sample of the band [39]. The effective acceleration is a combination of the response scatterers (human torso, arms, head or legs) isolating it from slow moving or stationary targets and electromagnetic interference signals from other sources (Wi-Fi) [41]. A weighted average of Doppler frequency shifts at each time step is used to calculate this acceleration. Then, a Gaussian filter is employed to focus on velocity ranges indicative of a fall, filtering out slower movements that could be associated with non-fall activities like sitting or standing. The choice of a band-pass filter over a high-pass filter minimizes interference from electronic noise and external signals which will display higher frequencies than expected. For example for a human fall, it is expected the maximum velocity to reach 5 m/s [42] while the acceleration will be around 10 m/s^2. In the STFT spectrum analysis, both positive (movement towards the radar) and negative (movement away from the radar) frequency bands are important, requiring the use of two

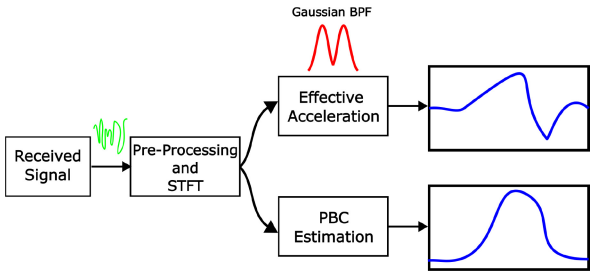


FIGURE 3. A Simple Flowchart of the estimation of the PBC and effective Acceleration and their indicative graphs for a fall case.

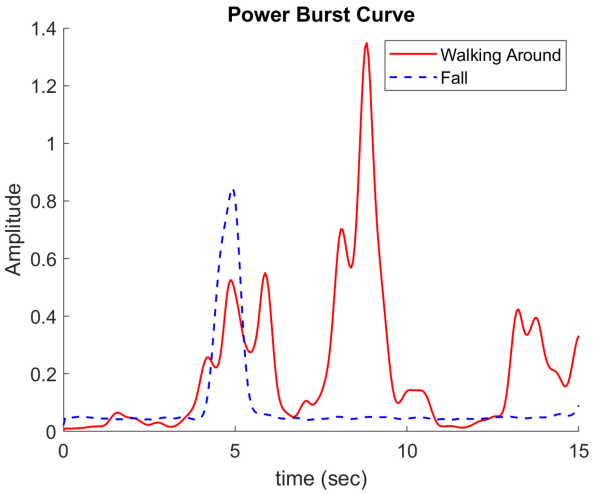


FIGURE 4. A Comparison between fall and non-fall case for PBC metric.

Gaussian distributions with the same variance but opposite mean values to effectively cover both scenarios. A simple flowchart of the above procedure is presented in Fig. 3. It was proved that this effective acceleration curve could be exploited to determine if a fall occurred or not with a SVM [24]. In Figs. 4 and 5, a comparison between a fall and a non-fall case (walking around the office) is taken place for the PBC and effective acceleration. As someone can observe the amplitude of the PBC curve cannot be used to determine the fall because the high power low frequency components contributes. However, when the acceleration is estimated the difference in fall and non-fall cases is more distinguishable. In this work, from the PBC and the effective acceleration, ten features will be extracted, as it will be analyzed in the next sections, and train different analog architectures in order to test their fall detection capabilities.

IV. DECISION TREE CLASSIFICATION MODEL

A Decision tree classification model is a fundamental machine learning algorithm used for both classification and regression tasks [43], [44], [45]. It operates by recursively partitioning the dataset based on feature values, creating a tree-like structure. The process starts at the root node, where the algorithm selects the most discriminative feature using metrics like Information Gain or Gini Impurity [46]. The chosen feature splits the data into subsets, each branch

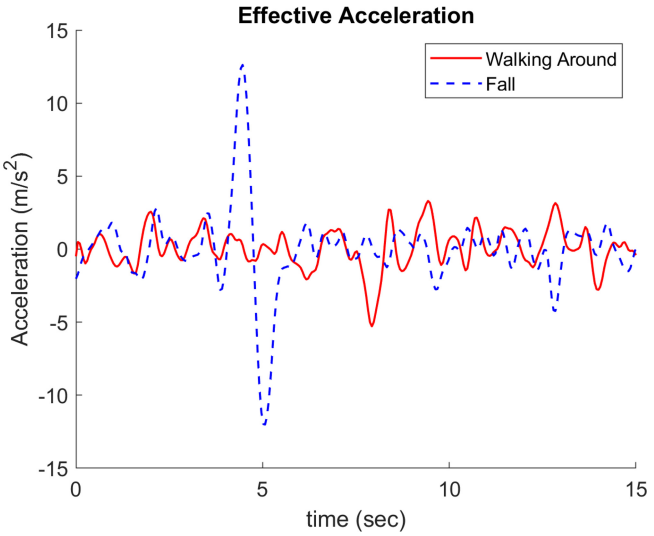


FIGURE 5. A Comparison between fall and non-fall case for the effective acceleration metric.

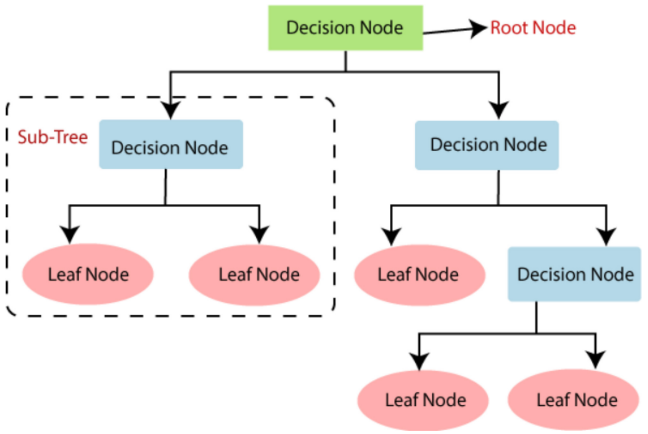


FIGURE 6. A block diagram of a Decision tree classification model. In a decision tree classification model, nodes can be of two types (decision node and leaf nodes). At each node, the algorithm selects the best feature and condition to split the data, aiming to maximize the purity of the resulting subsets with respect to the target variable. This process continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of samples per leaf node. The resulting tree can then be used to make predictions on new, unseen data by traversing the tree from the root node down to a leaf node.

representing a possible outcome. This process continues until the leaf nodes, which contain the predicted class labels. Mathematically, a Decision tree can be represented as a function $f(x)$ that takes an input vector x and maps it to a predicted class label or value. This can be expressed as:

$$f(x) = \sum_{i=1}^{N} c_i \cdot I(x \in R_i) \tag{1}$$

where N is the number of leaf nodes, c_i is the predicted class or value for the $i-th$ leaf node and R_i is the region of feature space associated with the $i-th$ leaf.

The split criteria in a Decision Tree Classification Model is determined using metrics that quantify the impurity or disorder of a dataset. For example, the Gini Impurity is commonly used and is defined as:

$$Gini(p) = 1 - \sum_{i=1}^{K} p_i^2 \tag{2}$$

where K is the number of classes and p_i represents the proportion of samples belonging to class i in the dataset. Another metric is Information Gain $IG(D, A)$, which measures the reduction in entropy after a split:

$$IG(D, A) = H(D) - \sum_{u} \frac{D_u}{D} H(D_u) \tag{3}$$

where $u \in Values(A)$, $Values(A)$ are the possible values of feature A, A is a candidate feature for splitting, $H(D)$ is the entropy of dataset D, D_u is the subset of D where feature A takes value u and $H(D_u)$ is the entropy of subset D_u. These equations illustrate the mathematical underpinnings of how a Decision Tree Classification Model makes decisions and performs splits based on impurity metrics.

The Decision tree model's effectiveness lies in its interpretability and ability to handle various types of data [45]. By visually representing the decision-making process, it offers valuable insights into the underlying relationships within the data. Moreover, Decision trees serve as building blocks for more complex ensemble methods like Random Forests and Gradient Boosted Trees. These ensembles combine multiple Decision trees to enhance predictive accuracy. By aggregating the predictions of individual trees, they can mitigate overfitting and improve generalization performance. This versatility, combined with the inherent transparency of Decision trees, makes them a valuable tool in machine learning across a wide range of applications.

A decision tree begins its classification process by starting from the root node, where it compares the attribute value of the dataset with that of the root [43], [44], [45]. Based on this comparison, it proceeds along the corresponding branch to the next node. In Fig. 6, a block diagram illustrates a Decision tree classification model which introduces key terminology. The root node serves node, on the one hand, as the starting point for the tree and represents the entire dataset, which is subsequently partitioned into two or more homogeneous sets. Leaf nodes, on the other hand, signify the final output and mark the endpoint of further segregation. Splitting involves the division of decision and root nodes into sub-nodes based on specified conditions. A sub-tree is formed by this process of splitting. On the contrary, pruning entails the removal of unnecessary branches from the tree. The root node is referred to as the parent node, while the remaining nodes are known as child nodes.

V. PROPOSED ARCHITECTURES AND MAIN BUILDING BLOCKS

In this section, three analog architectures of the Decision tree classification model, along with the fundamental circuits constituting it, are presented. More specifically, the proposed architectures approximate the behavior of the specific model

TABLE 1. Extracted features.

Square root of amplitude	$\text{SRA} = \left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{\lvert x_i\rvert}\right)^2$	**Impulse factor**	$\text{IF} = \frac{N\cdot\max(x_i)}{\sum_{i=1}^{N}\lvert x_i\rvert}$
Kurtosis value	$\text{KV} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu_x}{\sigma_x}\right)^4$	**Margin factor**	$\text{MF} = \frac{\max(x_i)}{\text{SRA}}$
Skewness value	$\text{SV} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{x_i-\mu_x}{\sigma_x}\right)^3$	**Frequency center**	$\text{FC} = \frac{1}{N}\sum_{i=1}^{N} f_i$
Peak-to-peak value	$\text{PPV} = \max(x_i) - \min(x_i)$	**Root-mean-square frequency**	$\text{RMSF} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} f_i^2}$
Shape factor	$\text{SF} = \frac{\max(x_i)}{\text{SV}}$	**Root variance frequency**	$\text{RVF} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - FC)^2}$

based on the classification process it employs. The conducted study focuses on the classification of data collected from the implemented Radar-Based system for detecting Human Falls. As a result, the classification task initially comprises 2 classes and 10 features. However, these architectures can be extended to accommodate N_{cla} classes and N_d features, allowing for greater generalization. The selected features (both type and number) are determined through feature extraction methods employed during the algorithm's training [47], [48], [49], [50]. This is done to attain maximum accuracy without encountering issues of overfitting. In Table 1 the 10 features are summarized. More specifically, they are selected based on their ability to capture important characteristics of signals, such as their amplitude distribution, frequency content, shape, and temporal dynamics. These features are commonly used in signal processing and are known to be informative in various applications such as pattern recognition, classification, and anomaly detection.

A. ANALYSIS OF MAIN BUILDING BLOCKS

The aforementioned architectures comprise varying combinations of the following three circuits: sigmoid function circuit (SFC) [31], [51], multiplier (MP) [52], and Winner-takes-All (WTA) circuit [53]. Both transistor-level implementations and the behavior of each circuit are provided. Referring to the first one, an alternative implementation of the classic SFC [31], [51] that generates a univariate sigmoid curve is introduced. Since its output is in the form of current, it can be easily expanded to higher dimensions by connecting additional SFCs to the same node. Concerning the analog MP here it employs the same implementation that was used to regulate the height of Gaussian curves in an analog Support Vector Machine (SVM) implementation [52] and is well-suited for small currents. As for the argmax operation, the standard Lazzaro WTA circuit [53] is employed. All of the mentioned circuits provide the necessary functionalities to achieve a high classification accuracy. Moreover, in order to minimize the system's power consumption, all transistors operate in the sub-threshold region [54] and the power supply rails are set to $V_{DD} = -V_{SS} = 0.3\text{V}$ across the entire architectures.

Since a low-power design is one of the main goals of this work, all transistors operate in the sub-threshold region, with power supply rails set to $V_{DD} = -V_{SS} = 0.3V$. The selection of the basic building blocks and power supply rails

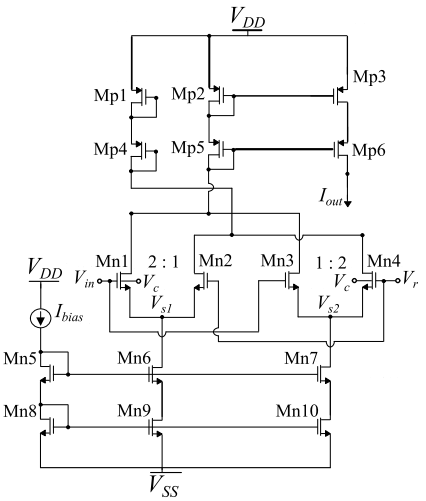


FIGURE 7. The analog hardware circuit related to the realization of sigmoid-based curves. It is a differential difference SFC with cascode current mirrors. The voltage V_{in} is related to the system's input. The voltage parameters V_r and V_c, along with the bias current I_{bias} tune the mean, variance and amplitude of the sigmoid function.

is guided by a trade-off between achieving high accuracy, minimizing power consumption, and ensuring the correct operating principles for the whole classifier. Also, we run noise-transient simulations to check the behavior of the proposed classifier. The classification result appeared to be quite robust. To a certain extend this tells us that the errors due to internal noise is small with respect to the errors in the data. Moreover, the ease of implementation of SFC, MP and WTA, makes them favorable candidates for area efficient and low-power classifiers.

1) SIGMOID FUNCTION CIRCUIT

Typical SFCs output a univariate sigmoid function curve. In this study a customized version of the SFC, which is illustrated in Fig. 7, is introduced and implemented to elevate the output curve's quality and enhance robustness. In particular, instead of a simple current mirror used in [31], [51], a cascode one ($M_{p2}, M_{p3}, M_{p5}, M_{p6}$) is selected (robust mirroring for small currents). Furthermore, for the purpose of adjusting the width of the output curve, a bulk-controlled differential difference pair (denoted as $M_{n1} - M_{n4}$) is employed in place of a basic one. In practice, this differential difference pair comprises two distinct differential pairs

TABLE 2. SFC's transistor dimensions.

NMOS	W/L (μ_m/μ_m)	PMOS	W/L (μ_m/μ_m)
M_{n1}, M_{n4}	1.6/0.4	M_{p1}-M_{p6}	1.6/1.6
M_{n2}–M_{n3}	0.8/0.4	-	-
M_{n5}–M_{n10}	0.4/1.6	-	-

capable of generating currents characterized by sigmoidal-shaped curves with adaptable slopes [55]. The transistors M_{n1} and M_{n4} are deep-n-well. Via the parameter voltage V_c and the body effect the output curve's width is tuned. In order to enhance the linearity of the block containing the differential difference pairs, the size ratio between transistors $M_{n1} - M_{n2}$ and $M_{n3} - M_{n4}$ is adjusted to 2, as opposed to the original ratio of 1 [56], [57].

All the circuit inputs and parameters affect the behavior of the output curve. The transistors M_{n1} and M_{n3} are connected to the input voltage, V_{in}, serving as conduits for information pertinent to the dataset. The electronic tuning of the sigmoid function's height, center and width is achieved through the manipulation of three circuit parameters: I_{bias}, V_r and V_c. These parameters are provided through the training process of the classifier (via a software-based implementation). It's worth noting that this calibration process is a one-time operation, after which the resultant parameters are extracted and stored in an analog memory system [58].

The bias current I_{bias}, depicted in Fig. 8, tunes the height of the resulting sigmoid output current under constant $V_r = 0$ and $V_c = -300mV$. The mean value of the derived sigmoid function is altered through the voltage V_r, as shown in Fig. 9, keeping the values of $I_{bias} = 5nA$ and $V_c = -300mV$ constant. The SFC's tunability in width, regulated by the parameter voltage V_c, is illustrated in Fig. 10 with fixed values of $I_{bias} = 5nA$ and $V_r = 0$. Elevating the parameter voltage V_c corresponds to an expansion of the sigmoid curve's width. Effective operation is achieved across a broad range of parameter voltage V_r, spanning from $V_{r_{min}} = -250mV$ to $V_{r_{max}} = 50mV$. This range is chosen because it is the effective range for the training procedure. For higher V_r values, a decrease in the output current is observed, which is no longer equal to the maximum but has a deviation. This is undesirable as it cannot achieve the desired separation between high and low currents for a small bias current I_{bias}. In this work, this circuit is employed as an approximated high-low output circuit. For $V_{in} > V_r$: I_{out} is equal to $2 \cdot I_{bias}$. For $V_{in} = V_r$: $I_{out} = I_{bias}$. For the rest, the output current is very close to zero. The SFC's transistor dimensions are summarized in Table 2.

2) ANALOG MULTIPLIER CIRCUIT

To achieve precise linear scaling, an analog multiplier circuit illustrated in Fig. 11 is employed. This multiplier (MP) operates on the translinear principle [31], [51], which stipulates that the product of clockwise translinear elements' currents in a translinear loop equals the product of counterclockwise translinear elements' currents derived within the same loop. Essentially, in sub-threshold region operating MOS, the translinear principle transforms the sum of gate-to-source voltages around the loop into a current

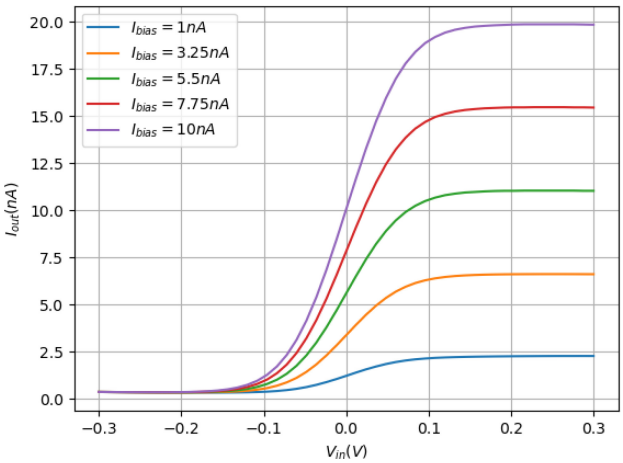


FIGURE 8. The output current of the SFC as a function of V_{in} and parameterized on I_{bias}, for $V_r = 0mV$ and $V_c = -300mV$.

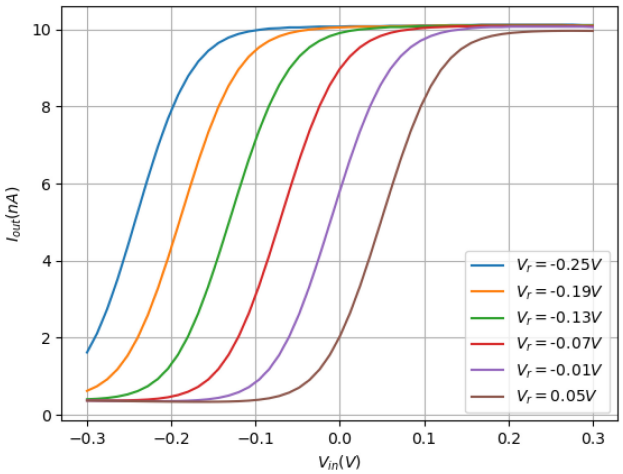


FIGURE 9. The output current of the SFC as a function of V_{in} and parameterized on V_r, for $I_{bias} = 5nA$ and $V_c = -300mV$.

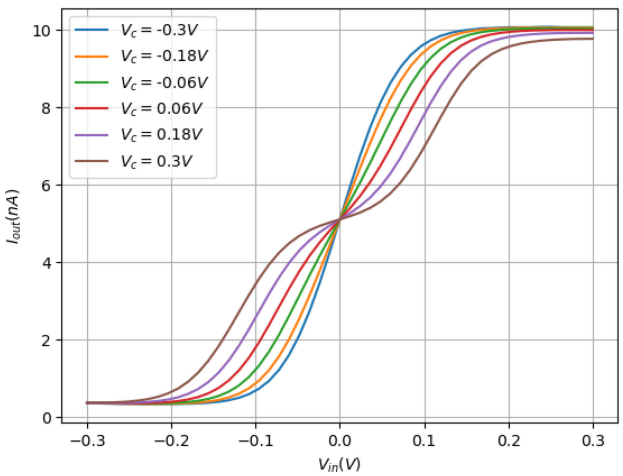


FIGURE 10. The output current of the SFC as a function of V_{in} and parameterized on V_c, for $V_r = 0mV$ and $I_{bias} = 5nA$.

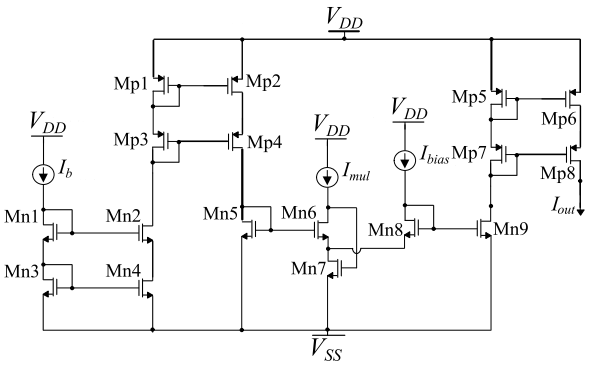


FIGURE 11. The analog translinear implementation of the multiplier. The three input currents are used to tune the output current based on a mathematical equation. An interesting observation is that when I_b is set to a high current value (more than $2nA$) and I_{bias} is set to a low current value (for example, $500pA$), the output current equals the low current value.

TABLE 3. Multiplier's transistor dimensions.

CMs	W/L (μ_m/μ_m)	TL	W/L (μ_m/μ_m)
M_{n1}–M_{n4}	0.4/1.6	M_{n5},M_{n9}	0.4/1.6
M_{p1}–M_{p4}	0.4/1.6	M_{n6}	3.6/1.6
M_{p5}–M_{p8}	0.4/1.6	M_{n8}	4/1.6
M_{n7}	1.2/0.8	-	-

product. This transformation is made possible due to the exponential characteristics of sub-threshold region operating MOS [54] current relative to its gate-to-source voltage, which stems from Kirchhoff's voltage law applied within the loop.

The designed translinear MP circuit features transistors M_{n5}, M_{n6}, M_{n8}, and M_{n9} arranged in an alternating loop topology [52]. This configuration ensures that the output current remains unaffected by the sub-threshold slope factor κ. Additionally, we incorporate cascode NMOS and PMOS current mirrors (consisting of transistors M_{n1}–M_{n4} and M_{p1}–M_{p8}, respectively) to achieve precise current mirroring. Assuming all four transistors (M_{n5}, M_{n6}, M_{n8}, and M_{n9}) are operating in the sub-threshold region and based on the translinear principle, the MP's output current can be expressed as:

$$I_{out} = \frac{I_b I_{bias}}{I_{mul}}. \tag{4}$$

In this case I_b and I_{bias} represent the inputs of the MP circuit and I_{mul} is a constant normalizing current. The inclusion of transistor M_{n7} is crucial for proper biasing of the translinear loop. Detailed dimensions of the transistors in the multiplier circuit are provided in Table 3.

Based on eq. (4), the output current I_{out} increases linearly with the increase of currents I_b and I_{bias} and decreases uniformly with the increase of current I_{mul}. The behavior of the circuit is also corroborated by the simulation results. The output current of the MP circuit as a function of I_{bias} and parameterized on I_{mull} for $I_b = 5nA$ is depicted in Fig. 12. The output current of the MP circuit as a function of I_{bias} and parameterized on I_b for $I_{mul} = 10nA$ is shown in Fig. 13. The output current of the MP circuit as a function

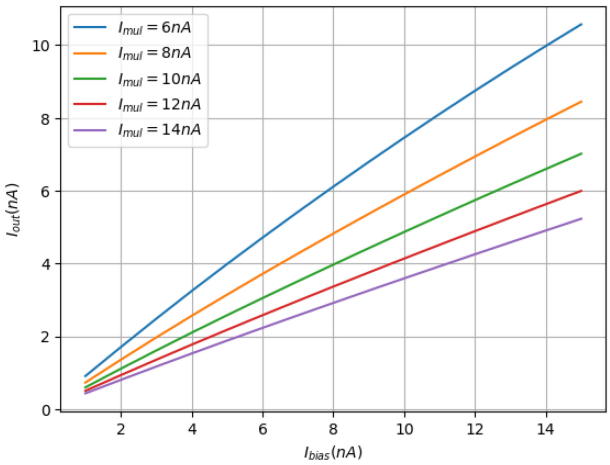


FIGURE 12. The output current of the MP circuit as a function of I_{bias} and parameterized on I_{mull} for $I_b = 5nA$.

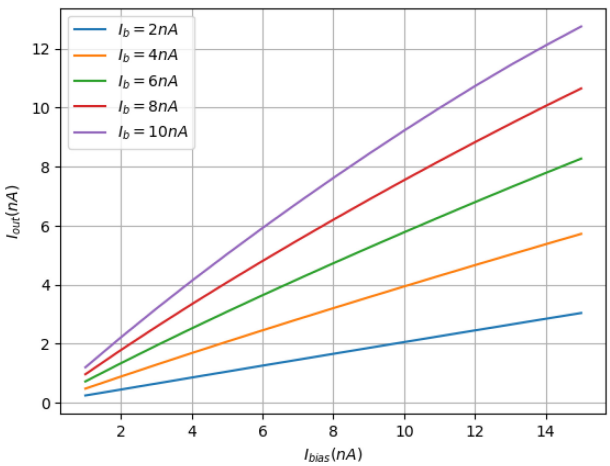


FIGURE 13. The output current of the MP circuit as a function of I_{bias} and parameterized on I_b for $I_{mul} = 10nA$.

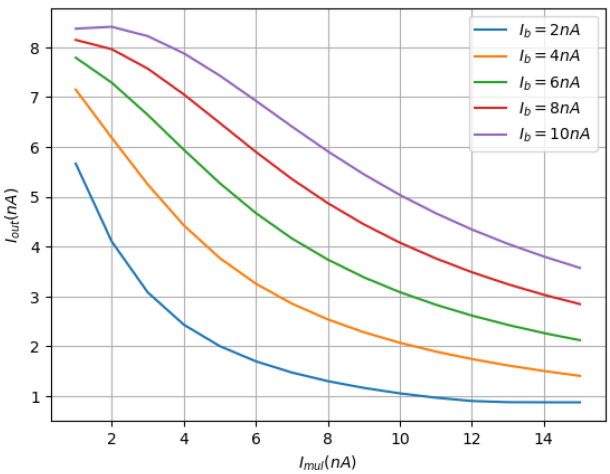


FIGURE 14. The output current of the MP circuit as a function of I_{mul} and parameterized on I_b for $I_{bias} = 5nA$.

of I_{mul} and parameterized on I_b for $I_{bias} = 5nA$ is depicted in Fig. 14. An interesting observation is that when I_b is set to a high current value (more than $2nA$) and I_{bias} is set to a

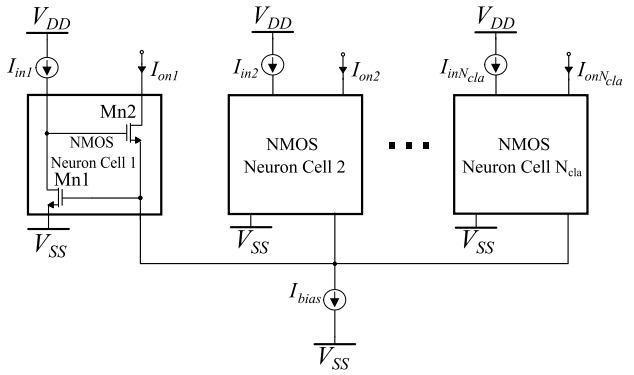


FIGURE 15. The conventional Lazzaro NMOS-based argmax operator (WTA) structure involves N_{cla} neurons.

low current value (for example $500pA$), the output current equals the low current value. More specifically, if one value is $I_b = 5nA$ (high) and the other value is $I_{bias} = 500pA$ (low), then the circuit approximates an "AND Logic Gate" and the output is equal to $500pA$ (low) and not equal to $2.5nA$ (median). Based on this behavior, the implementation of a hardware approximation of the Decision tree classifier becomes straightforward. Specifically, even if there are data that do not belong to the class that is studied, this will be verified by all the features of the classifier. In the case of two correlated features, the following occurs: one may output high and the other may output low. Therefore, their multiplication must result in a low value, since the data do not belong to the class. If the function was different, the output would occur at some intermediate value and thus the classifier could assume that the data belong to the class.

3) WINNER-TAKES-ALL CIRCUIT

The Winner-Takes-All (WTA) circuit is designed to process a set of N_{cla} input signals, responding exclusively to the largest input while suppressing the responses from the remaining $N_{cla} - 1$ inputs. Essentially, the WTA circuit embodies the *argmax* function. Over time, various implementations of WTA circuits have emerged, including voltage-mode designs [53], current-mode configurations [59] and even an ultra-low supply voltage version operating at just $0.3V$ [60]. It's worth noting that all current-mode WTA circuit architectures are derived from the original design introduced by Lazzaro [53].

The circuit configurations of the NMOS WTA circuit for N_{cla} inputs is illustrated in Fig. 15. The PMOS WTA circuit is the symmetric one and it can easily designed. In the simplest NMOS version, a straightforward 2-input WTA circuit ($N_{cla} = 2$) is constructed using four NMOS transistors with identical W and L parameters, operating in the sub-threshold region and biased by a constant current, denoted as I_{bias}. The dimensions of these transistors are specified as $(W/L) = \frac{400nm}{1600nm}$. When the input currents are equal, i.e., $I_{in1} = I_{in2}$, the output currents become $I_{on1} = I_{on2} = 0.5I_{bias}$. For this specific work, we have not observed equal input currents during classification procedure. As a results, we

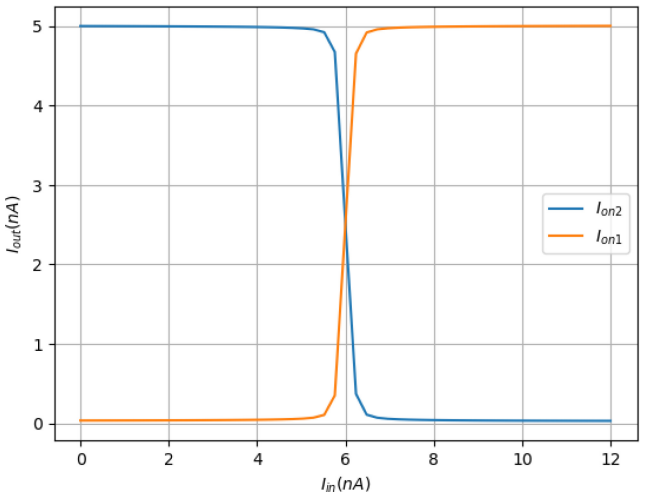


FIGURE 16. In this configuration, $I_{bias} = 5nA$, I_{in1} is a parametric current equal to I_{in} and $I_{in2} = 6nA$. The output current for both neurons as function of the input current I_{in1}.

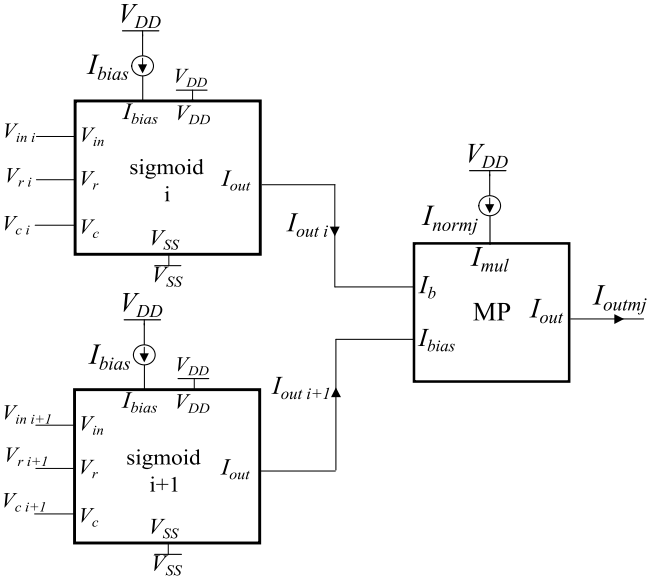


FIGURE 17. The AFB block which consists of two correlated features which are represented by two SFCs and a MP circuit.

have not multiple winners. For this reason, modifications to the WTA or alternative approaches such as cascaded WTA are not necessary. Since both M_{n1} for class 1 ($Mn1cl1$) and M_{n1} for class 2 ($Mn1cl2$) have the same V_{GS} voltage, when $I_{in1} > I_{in2}$, it follows that $V_{D_{Mn1cl1}} = V_{G_{Mn2cl1}} > V_{G_{Mn1cl2}} = V_{D_{Mn2cl2}}$ (same symbols for M_{n2} transistor). Assuming that both output transistors M_{n2cl1} (M_{n2} class 1) and M_{n2cl2} (M_{n2} class 2) operate in deep sub-threshold and have the same source voltage, a slight disparity in their gate voltages leads to a disproportionately larger difference in the output currents. In this scenario, $I_{on1} = I_{bias}$ and $I_{on2} = 0$. Consequently, for input currents that differ significantly, only the output current corresponding to the highest input current will register a non-zero value. The behavior of the WTA circuit is depicted in Fig. 16. Here, $I_{bias} = 5nA$, I_{in1} is a parametric current equal to I_{in} and $I_{in2} = 6nA$.

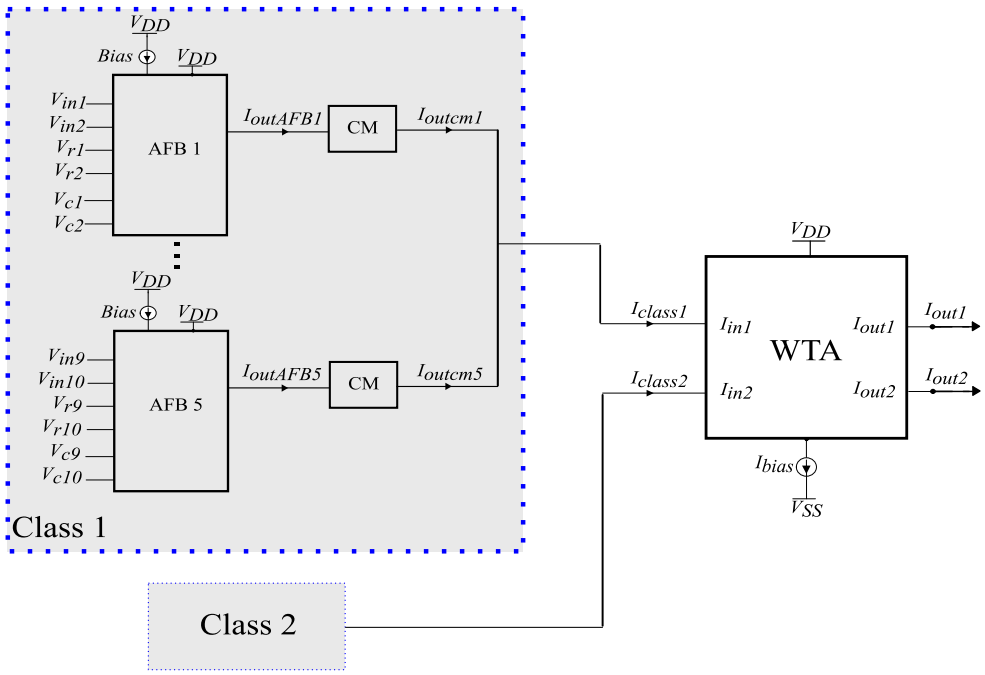


FIGURE 18. The high level architecture of a Decision tree classification model implemented with 5 AFBs, 5 CMs for each class and a WTA circuit. This architecture is called SCA.

B. PROPOSED ANALOG CLASSIFIER ARCHITECTURES

In this subsection the proposed classifier's high level architectures are discussed. Since it is an application specific task, the proposed architectures consists of $N_{cla} = 2$ classes and $N_d = 10$ input dimensions. Although all of these architectures are application-specific, we can readily create a generalized version to accommodate more or fewer classes and input features. The implemented architectures are hardware-friendly approximations of the Decision tree classification model.

1) SIGMOID CORRELATED ARCHITECTURE

Initially, prior to implementing the architecture, a test was conducted to assess the correlation between each pair of selected features. Once the correlated features (pairs) are identified, the hardware architecture is then designed. The training process is conducted in the software, after which the parameter values are stored in analog memories [58]. Afterwards, the hardware checks for each feature whether the input data belong to a class or not. If the input data belong to the class, the output of each feature should be high. There are cases where the output is high for a feature, but these input data do not belong to the class (random case). If each correlated feature has a low output, then the final output will be low for that class, considering those two features. This process is executed for all five pairs of correlated features for both classes.

Initially, each feature in this work is represented by a sigmoid function. In the previous subsection the behavior of the proposed SFC is analysed. A MP circuit is employed to implement the correlation between the features. This implementation is based on the following principle: if the output of one feature is high and the output of the correlated one is low, then the final output is low. The circuit implementation that models the behavior of the two correlated features is illustrated by the architecture shown in Fig. 17. For simplicity this realization is called AFB block. This is a generalized schematic which combines the two activation function circuits, which are related to feature i and the correlated one $i + 1$ (for $i = 1, 3, 5, 7, 9$) and the correlated circuit (MP). Each SFC receives an input voltage V_{ini} or V_{ini+1}, along with parameter voltages and currents obtained from the training procedure and produces a high or low output current referred as I_{outi} or I_{outi+1}. Then, the two output currents are correlated through the MP and the correlated output current for the two features I_{outmj} is extracted ($j = 1, 2, 3, 4, 5$).

The high-level structure of the sigmoid correlated architecture encompasses 2 classes and 10 features. Each block associated with a class is composed of 5 AFBs, each yielding an output current denoted as $I_{outAFBj} = I_{outmj}$ ($j = 1, 2, 3, 4, 5$). Additionally, there are 5 cascode current mirrors (CMs) in place to mitigate potential distortions. The summation of the CMs' currents generates the class current, which can be either I_{class1} or I_{class2}. In the final stage, the WTA circuit is employed to compare the two class currents and determine the final decision.

2) SIGMOID DECISION ARCHITECTURE

In this architecture (Sigmoid decision) there is no requirement for correlated features. Given that each feature operates

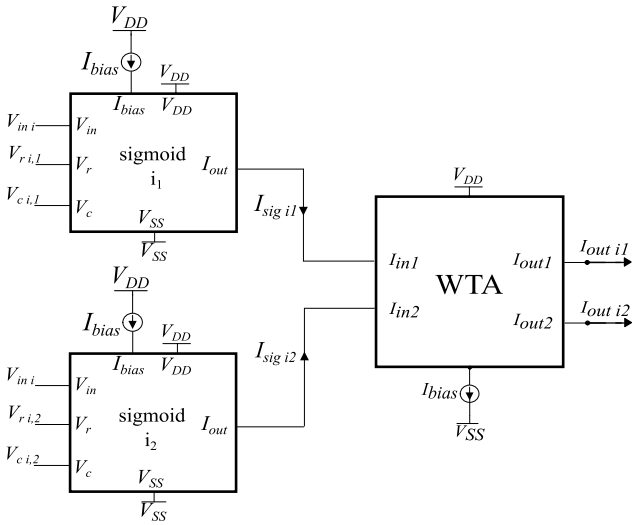


FIGURE 19. The SDC block which consists of two SFCs which represent the same feature and an argmax operator (WTA) circuit which compares both SFCs' outputs.

independently, we can straightforwardly assess the probabilities of each feature for each class, allowing us to identify the highest probability. If an input datum is related to a specific feature and belongs to this class, then the output current is high, else it is low. The circuit implementation that models this behavior is depicted in Fig. 19. For simplicity this realization is called SDC block. This is a generalized schematic that integrates the two SFCs, which are related to feature i (for $i = 1, 2, .., 9, 10$) and more specifically i_1 (class 1) and i_2 (class 2) along with the current comparator circuit (WTA). Each of these circuits receives the input data. If it belongs to the class, the output current I_{sigi} of the respective SFC is high; otherwise, it is low. There are cases in which both currents have high value. To address this, the WTA circuit compares the two currents (I_{sigi}) and determines the final high and low currents for each specific feature i.

The high-level structure of the sigmoid decision architecture again encompasses 2 classes and 10 features. Each block associated with a class is comprised of 10 SDCs' output currents, each one referred as $I_{outi,j}$ (for $i = 1, 2, .., 9, 10$ and $j = 1, 2$). Additionally, there are 20 cascode CMs in place to reduce potential distortions, with each one corresponding to a WTA's output. The summation of the CMs' currents generates the class current, which can be either I_{class1} or I_{class2}. In the final stage, the WTA circuit is employed to compare the two class currents and determine the final decision.

3) SIGMOID THRESHOLD ARCHITECTURE

In this architecture (Sigmoid threshold) the same test was also carried out to evaluate the correlation between each pair of selected features. The training phase takes place in the software, where parameter values are subsequently stored in analog memories [58]. Following this, the hardware investigates each feature to determine if the input data correspond to a specific class. In case the input data align with the class, the output for each feature is expected to register as high. Same as the other architectures, there are cases where the output is high for a feature, but this input data do not belong to the class. In such cases, if the correlated feature exhibits a low output, the final output for that class will also be low, accounting for these two features. This meticulous process is iterated for all five pairs of correlated features for both classes.

In the current architecture, a distinct design path is pursued to address situations where a feature produces a high output as a result of a random event. More specifically, the output current of an SFC can assume an intermediate value, which is not considered low. This may provide an additional "vote" for the class to which the input data do not belong. The solution to this case is the addition of an extra WTA in the output of each SFC. As a result the two correlated features (i and $i + 1$ for $i = 1, 3, 5, 7, 9$) consist of two SFC, two WTA circuits and a MP circuit. This building block diagram is illustrated in Fig. 21 and for simplicity this block is called STC. Each SFC receives an input voltage V_{ini} or V_{ini+1}, along with parameter voltages and currents obtained from the training procedure and produces an output current referred as I_{sigi} or I_{sigi+1}. Then, each of the currents is compared with a threshold current (I_{thi} or I_{thi+1}) via a WTA circuit and provide a "checked" high or low current I_{fei} or I_{fei+1}. Then, the two output currents are correlated through the MP and the output current of the correlated features I_{outmj} is extracted ($j = 1, 2, 3, 4, 5$).

The high-level structure of the sigmoid threshold architecture is shown in Fig. 22. This architecture consists of 5 STCs, 5 current mirrors, one WTA circuit and a threshold current. In this case only the probability of one class is calculated (if the input data belong or not to this class). All the 5 I_{outmj} output currents are summed through 5 CMs (to reduce potential distortions). Then the current related to the specific class I_{class} is compared with a specific threshold current (I_{thc}). This procedure determines the final winner. If $I_{class} > I_{thc}$, then class 1 emerges as the winner; otherwise, it is class 2.

VI. OFFLINE TRAINING AND ARCHITECTURE TUNABILITY

The analog hardware architectures of Decision tree classification model outlined above rely on the integration of the SFC to serve as a distance metric for prototypes in each class (as detailed in previous Section). This setup allows for the electronically adjustable parameters, V_r and V_c, to be leveraged in creating a post-layout classification chip. Also, the related current can be tuned in all SFC, MP and WTA blocks in order to provide higher tolerance at the cost of power consumption (for more complex tasks). This adaptability enables easy customization to suit the specific requirements of the targeted application. Additionally, the system's tunability is able to address a diverse array of classification challenges, regardless of factors such as input dimensions (N_d) and the number of classes (N_{cla}).

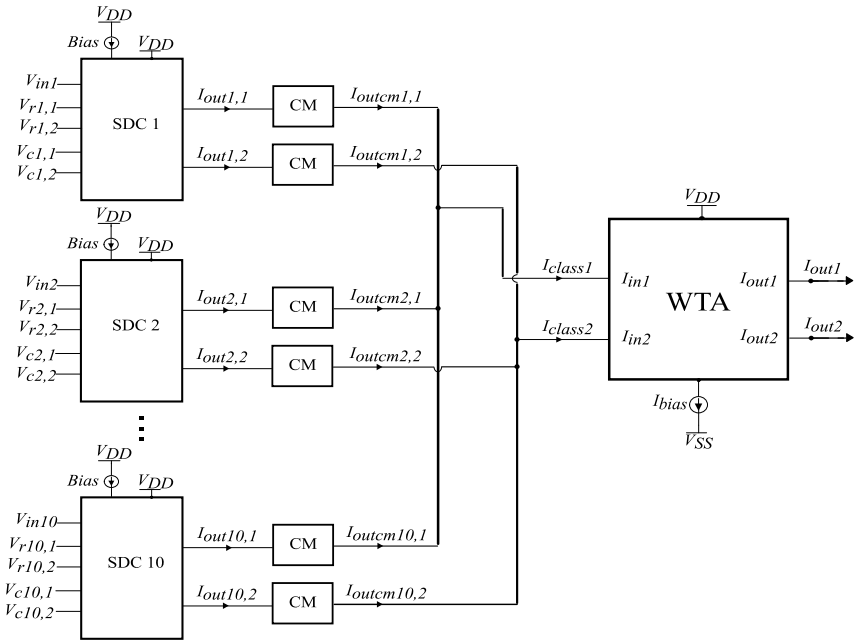


FIGURE 20. The high level architecture of a Decision tree classification model implemented with 10 SDCs, 20 CMs and a WTA circuit. This architecture is called SDA.

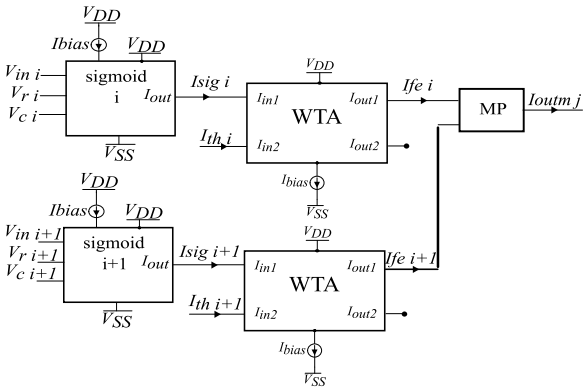


FIGURE 21. The STC block which consists of two correlated features which are represented by two SFCs, two WTAs (compare the SFC with a threshold current) and a MP circuit.

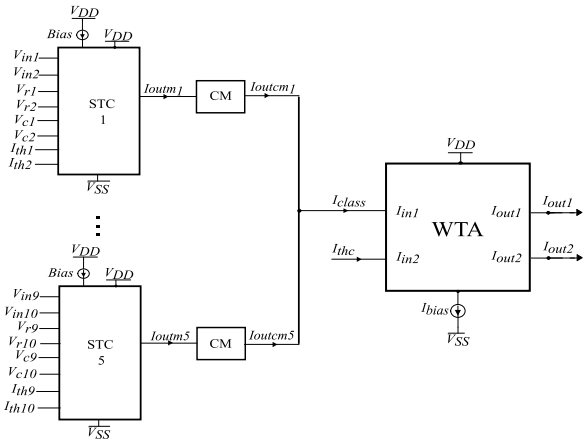


FIGURE 22. The high level architecture of a Decision tree classification model implemented with 5 STCs, 5 CMs and a WTA circuit. A threshold current is used in order to provide the appropriate decision boundary. This architecture is called STA.

To initiate the process, we developed a software-based implementation of the Decision tree procedure to gather essential parameters for the circuit. To ensure consistency, all datasets used for validating the classifier were normalized to fall within the operational range of the implemented SFC, specifically within the range of $[-250, 50]$mV, as detailed in previous Section. Also, regarding the range of the V_c is set to $[-300, 300]$mV. This provides an extra degree of freedom in the implementation, because the parameter voltage V_c is able to tune the variance of the curve. As a result, a map between the values of the V_c and the related variance is implemented. Following this, the software-based classifier underwent a tailored training process using a specific methodology. This method enabled the extraction of input dimensions for each class, which directly correspond to the voltage parameters (V_r and V_c) of the hardware counterpart.

To obtain the values of I_{bias}, a deliberate choice was made due to the absence of a direct method within the Decision tree algorithm to ascertain them through the training process. It was decided to assign it an arbitrary value that remained constant across both classes. The selection of the I_{bias} is related to a trade-off between accuracy and power-consumption. This deliberate decision serves to pinpoint any notable decrease in accuracy in the hardware implementation to the extraction of software-based V_r, V_c values, simplifying the development process and minimizing unnecessary complexity. This step is executed once for each distinct application and the resulting parameters are subsequently exported and stored in an analog memory [58].

Training a decision tree classification algorithm involves several steps [43], [44], [45]. In the following a high-level

overview of the process is analysed step by step. 1) First, the collection of a labeled dataset is necessary. 2) Then, the dataset is divided into two subsets: a training set and a testing set. The training set is used to train the model, while the testing set is used to evaluate its performance. 3) The decision of the relevant features (attributes) is necessary for the classification task. There are features which may not contribute much to the decision-making process. 4) The decision tree algorithm recursively splits the data based on the selected features to create a tree-like structure. 5) Choose a splitting criterion. Common ones include Gini impurity and entropy (both are used for comparison and provide same results). These measure the impurity or disorder in a set of examples. 6) Starting from the root node, the best feature should be chosen to split the data. This is done by evaluating the splitting criterion for each possible split. 7) Stopping criteria are defined to determine when to stop growing the tree. This could be based on the depth of the tree, the number of examples in a node, or other factors. 8) After the tree is fully grown, pruning techniques are applied to reduce its size and complexity. This can help prevent overfitting. 9) The trained tree is used to make predictions on new data. Starting at the root node and following the branches based on the feature values until a leaf node is reached, which corresponds to a class label. 10) The testing set is used to assess the performance of the trained model.

In this work, each feature is independent from the others, except from the correlated ones. As a result, if a generalized implementation is provided, it can easily tune the number of input dimensions and classes. In the case of a generalized implementation with N_{cla} classes, it is possible to deactivate $N_{cla} - 2$ classes either by biasing them (each feature) with zero I_{bias} currents or by providing V_{in} values far away from V_r (for example, setting $V_{in} = V_{DD}$ and $V_r = V_{SS}$). Also, in the case of an implementation with N_d features, it is possible to deactivate $N_d - 1$ input features either by biasing each SFC with zero I_{bias} currents or by providing V_{in} values far away from V_r (for example, setting $V_{in} = V_{DD}$ and $V_r = V_{SS}$). Also, we can easily deactivate the whole classifier by using the previous technique for all classes and input dimensions.

VII. SIMULATION RESULTS

In this Section, to demonstrate the proposed architecture's proper operation, the classifier is tested on the collected dataset (fall-detection method). The related dataset and fall-detection methodology is analysed in Section II. The data are separated into a training and a test set ($70 - 30\%$ training-test split). The design process was executed using the Cadence IC suite within a TSMC 90 nm CMOS process. The simulation results were obtained on the layout for each specific architecture. These layouts, which are depicted in Figs. 23, 24, 25, incorporating the three decision tree classifiers, is meticulously crafted with a primary focus on achieving area efficiency. All layouts are implemented based on the common-centroid technique and additional dummy transistors are added to mitigate mismatches and address manufacturing considerations.

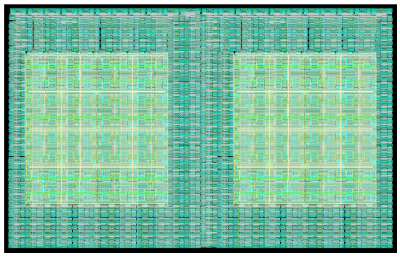


FIGURE 23. Layout of the proposed classifier architecture (SCA). The total area is equal to $0.351 mm^2$.

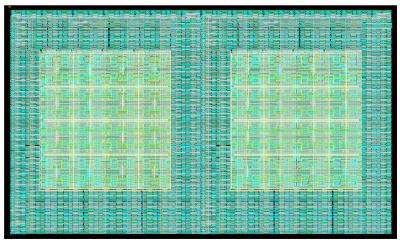


FIGURE 24. Layout of the proposed classifier architecture (SDA). The total area is equal to $0.413 mm^2$.

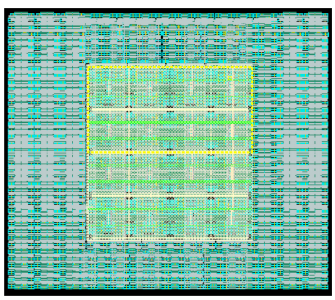


FIGURE 25. Layout of the proposed classifier architecture (STA). The total area is equal to $0.198 mm^2$.

In order to demonstrate the effectiveness of the proposed classification pipeline, we meticulously carry out a comprehensive training and validation process. This process is iterated 20 times for this specific dataset (for each architecture) to establish a robust classification accuracy, mitigating the impact of random variables inherent in software-based train-test divisions. In every iteration a direct comparison between the analog and software implementations is conducted, employing identical training and validation datasets. Furthermore, a sensitivity analysis of the circuit is executed through Monte Carlo simulations, encompassing $N = 100$ data points.

The behavior of the proposed analog classifiers is confirmed via the simulation results. In Table 4 the classification accuracy for the three classifiers is summarized. For the sigmoid correlated architecture, here called SCA, the hardware-based implementation of the proposed classifier exhibits a marginally lower accuracy of approximately 0.2% compared to an identical software-based counterpart. Furthermore, their respective results demonstrate similar

TABLE 4. Accuracy results on the dataset (over 20 iterations).

Approach	Best (%)	Worst (%)	Mean (%)
Software	99.60	96.30	98.04
SCA	99.40	96.10	97.84
SDA	98.00	94.70	96.44
STA	98.30	95.00	96.74

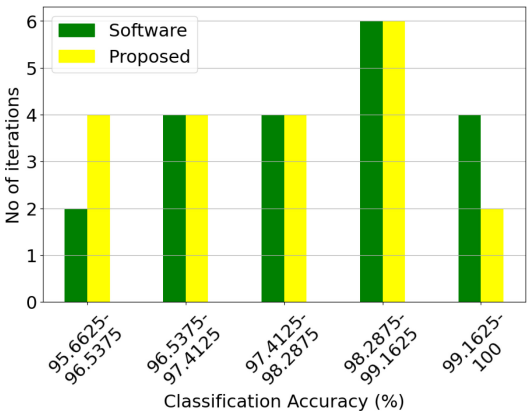


FIGURE 26. Classification results of the proposed SCA (yellow) and the equivalent software model (green) on the dataset over 20 iterations.

deviations across various train-test iterations. For a comprehensive comparative analysis, the results are depicted in Fig. 26, which provides detailed histograms depicting the exact classification accuracy. In the case of the sigmoid decision architecture (referred to as SDA), the hardware-based implementation of the suggested classifier shows a slightly lower accuracy, approximately 1.6%, compared to its software-based equivalent. Additionally, both implementations display similar deviations in their results across different train-test iterations. For a thorough comparative assessment, the results are presented in Fig. 27, which includes detailed histograms illustrating the precise classification accuracy. In the case of the sigmoid threshold architecture, referred to as STA, the hardware-driven implementation of the suggested classifier shows a slight drop in accuracy, approximately 1.3% lower than its software-based counterpart. Moreover, both implementations yield comparable deviations in their results across different train-test iterations. For a comprehensive comparative analysis, the results are illustrated in Fig. 28, which includes detailed histograms showcasing the precise classification accuracy.

The sensitivity behavior of each classifier circuit is verified via the Monte Carlo analysis. In this evaluation, the training data of one of the 20 candidates from the preceding test was employed as input. The outcomes are visualized through the Monte Carlo histogram presented in Figs. 29, 30, 31 for each classifier respectively. The mean value stands at $\mu_M = 98.079\%$, $\mu_M = 98.761\%$, $\mu_M = 98.431\%$ demonstrating close proximity to the mean value of the previous test, respectively. Additionally, the standard deviation is impressively low, recorded at $\sigma_M = 0.54\%$,

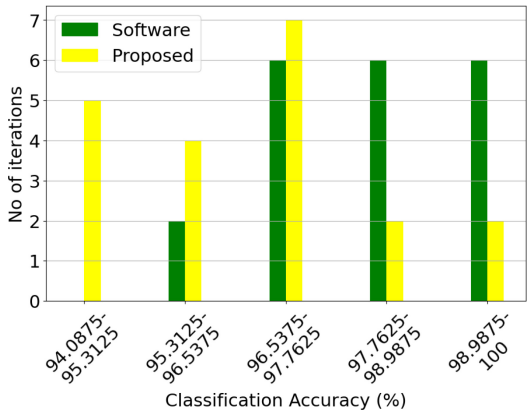


FIGURE 27. Classification results of the proposed SDA (yellow) and the equivalent software model (green) on the dataset over 20 iterations.

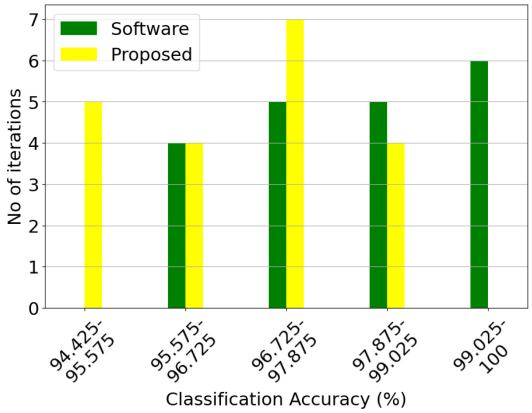


FIGURE 28. Classification results of the proposed STA (yellow) and the equivalent software model (green) on the dataset over 20 iterations.

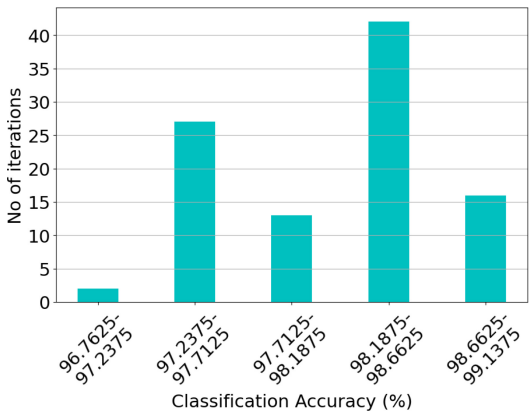


FIGURE 29. Post-layout Monte-Carlo simulation results of the proposed SCA on the dataset.

$\sigma_M = 0.59\%$, $\sigma_M = 0.61\%$, respectively. The previous values confirm that all three classifiers have high.

Apart from Monte-Carlo analysis, the proposed classifiers undergo testing to account for Process-Voltage-Temperature (PVT) variations. The selected corners encompass TT, SS, FF, SF, FS (T: Typical, S: Slow, F: Fast). Additionally, the power supply rails fluctuate within the range $V_{DD} = -V_{SS} = 0.25V$ to $V_{DD} = -V_{SS} = 0.35V$. Regarding

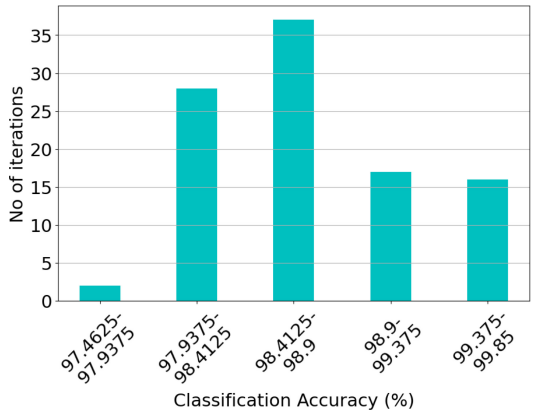


FIGURE 30. Post-layout Monte-Carlo simulation results of the proposed SDA on the dataset.

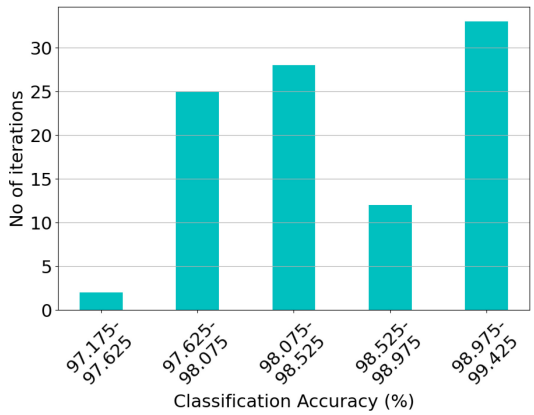


FIGURE 31. Post-layout Monte-Carlo simulation results of the proposed STA on the dataset.

temperature, the assessed spectrum spans from -25^oC to 125^oC. All three implementation exhibit resilience across corners, maintaining a minimum classification accuracy of 94.53%, 93.12%, and 93.46% for the SCA, SDA, and STA architectures, respectively, under the worst-case scenario.

The most challenging corner scenario emerges with SS, -25^oC, $V_{DD} = -V_{SS} = 0.25V$, coupled with reduced software-based accuracy (worst case). Specifically, in sub-threshold, devices should be biased with V_{GS} voltages almost equal to V_{th} (which increases with decreasing temperature due to the increase in carrier mobility) and in the slow corner, the doping concentration may be lower, leading to a higher V_{th} [54], [55], [61], [62]. As a results higher V_{GS} voltages are necessary. Also, in this region $V_{DS} \geq 4V_T$ where $V_T = kT/q$ (temperature-dependent). For a minimum temperature of -25^o Celsius, the value of V_T is 20.67mV. In this case, we may encounter a mismatch during mirroring due to the Early effect. We can address this issue by using cascode CMs and transistors with longer channel lengths (L).

VIII. COMPARISON STUDY AND DISCUSSION

In the related literature, it is notable that most analog classifiers are typically tailored to specific applications. This specificity poses a challenge when attempting to conduct an unbiased comparison among diverse implementations. As a result, there is an opportunity to customize the design of related classifiers to suit the same application, enabling a comprehensive performance evaluation across various ML models and approaches. All the summarized classifiers are implemented in a TSMC 90 nm CMOS process technology, with power supply rails selected based on the operating region and a trade-off between higher accuracy and lower power consumption. All are implemented for the same dataset.

All classifiers were trained using the required software, which relied on the mathematical models described in each implementation. Subsequently, they were all designed using the TSMC 90nm CMOS process. At this stage, they underwent schematic-level verification (except our work, which is verified in layout-level too), and necessary enhancements were implemented to optimize classification accuracy and speed while prioritizing minimal power consumption. We followed the necessary design process described in each work. In cases where the architecture operate in saturation, we applied the corresponding techniques specific to that operating region. The aforementioned process aimed to ensure a fair comparison, given that the implementations were carried out using different technologies and for distinct classification tasks.

Specifically, Table 5 offers a comprehensive overview of our research in comparison to related classifiers, including a SVM [52], [63], a cascaded-connected Bayes [64], a Gaussian mixture model (GMM) [65], a Radial Basis Function (RBF) [66], a RBF-Neural Network (NN) [67], a Multilayer Perceptron (MLP) [68], a K-means [69], a Support Vector Regression (SVR) [70], a Self-Organized Map (SOM) [71], a Long Short-Term Memory (LSTM) [72], a Fuzzy [73], a Threshold [74] and a cascaded-connected Centroid classifier [75], within the context of fall detection method. For further insights into analog and mixed signal classifiers, please consult [76], where these classifiers are summarized and elucidated.

All the implementations of ML models referenced in Table 5 are based on approximations of equivalent mathematical models. When it comes to architectural complexity, there are various approaches, some with low complexity and others with high complexity. The level of complexity is also tied to the specific ML model being implemented and the nature of the approximation. Furthermore, in contrast to cascaded implementations, each feature in this work is independent of the uncorrelated features. This leads to a higher classification accuracy. Additionally, in a more generic architecture, these implementations can handle a higher number of input dimensions compared to the cascaded approach. This provides a notable advantage by eliminating the need for Principal Component Analysis (PCA) [77].

Among the architectures examined in this study, namely, SCA, SDA, and STA, the SCA emerges as the top performer in achieving classification accuracy. This superiority is attributed to the quality of the SCA architecture's approximation compared to the other

TABLE 5. Analog classifiers' comparison on the dataset.

	Classifier	Min accuracy (%)	Mean accuracy (%)	Max accuracy (%)	Power consumption	Processing speed $\left(\frac{\text{classifications}}{\text{s}}\right)$	Energy (pJ) per classification	Architectures Complexity
This work	SCA	96.10	97.84	99.40	791 nW	$270K\frac{\text{classifications}}{\text{s}}$	$\frac{2.93\text{ pJ}}{\text{classification}}$	Medium
This work	SDA	94.70	96.44	98.00	598 nW	$250K\frac{\text{classifications}}{\text{s}}$	$\frac{2.392\text{ pJ}}{\text{classification}}$	Medium
This work	STA	95.00	96.74	98.30	495 nW	$310K\frac{\text{classifications}}{\text{s}}$	$\frac{1.59\text{ pJ}}{\text{classification}}$	Medium
[52]	SVM	93.10	95.87	98.10	65.3 μW	$140K\frac{\text{classifications}}{\text{s}}$	$\frac{466.43\text{ pJ}}{\text{classification}}$	Medium
[63]	SVM	95.10	97.45	98.90	654.32 μW	$870K\frac{\text{classifications}}{\text{s}}$	$\frac{751.95\text{ pJ}}{\text{classification}}$	High
[64]	Bayes	92.60	94.12	96.10	690 nW	$100K\frac{\text{classifications}}{\text{s}}$	$\frac{6.90\text{ pJ}}{\text{classification}}$	Low
[65]	GMM	93.10	95.39	97.20	1.46 μW	$100K\frac{\text{classifications}}{\text{s}}$	$\frac{14.60\text{ pJ}}{\text{classification}}$	Medium
[66]	RBF	91.20	93.33	94.10	23.12 μW	$120K\frac{\text{classifications}}{\text{s}}$	$\frac{192.67\text{ pJ}}{\text{classification}}$	Medium
[67]	RBF-NN	92.40	95.17	96.80	1.28 μW	$210K\frac{\text{classifications}}{\text{s}}$	$\frac{6.09\text{ pJ}}{\text{classification}}$	Medium
[68]	MLP	96.20	97.41	99.80	793μW	$930K\frac{\text{classifications}}{\text{s}}$	$\frac{852.69\text{ pJ}}{\text{classification}}$	High
[69]	K-means	96.40	97.12	99.60	245.2 μW	$5M\frac{\text{classifications}}{\text{s}}$	$\frac{49.04\text{ pJ}}{\text{classification}}$	High
[70]	SVR	94.70	95.81	96.70	83.1 μW	$870K\frac{\text{classifications}}{\text{s}}$	$\frac{95.51\text{ pJ}}{\text{classification}}$	High
[71]	SOM	93.50	95.81	97.80	696.8 μW	$180K\frac{\text{classifications}}{\text{s}}$	$\frac{3.87\text{ nJ}}{\text{classification}}$	Medium
[72]	LSTM	98.30	99.70	100.00	51 mW	$870M\frac{\text{classifications}}{\text{s}}$	$\frac{58.62\text{ pJ}}{\text{classification}}$	Very High
[73]	Fuzzy	93.20	95.43	97.40	920 nW	$4.55K\frac{\text{classifications}}{\text{s}}$	$\frac{202.19\text{ pJ}}{\text{classification}}$	Medium
[74]	Threshold	93.70	96.12	97.80	321 nW	$100K\frac{\text{classifications}}{\text{s}}$	$\frac{3.21\text{ pJ}}{\text{classification}}$	Low
[75]	Centroid	92.40	96.74	98.90	2.53 μW	$170K\frac{\text{classifications}}{\text{s}}$	$\frac{18.07\text{ pJ}}{\text{classification}}$	Medium

approaches. Additionally, this implementation surpasses all the other classifiers listed in Table 5 in terms of mean accuracy, with the exception of the LSTM algorithm, which excels in balancing model complexity and hardware-approximation efficiency. It's important to highlight that this heightened performance is attained with the least energy per classification in comparison to alternative approaches. While the Threshold classifier achieves the lowest power consumption, it does so at the expense of accuracy and processing speed, owing to its model's simplicity. To conclude, this work provides a trade-off between power-consumption, energy per classification and classification accuracy. It is important to highlight that, in this kind of biomedical applications, rapid processing speed is not a vital specification (sacrifice speed for power consumption).

IX. CONCLUSION

A Radar-based system for detection of human fall was introduced. The implemented fall-detection method was tested utilizing a low cost CW Radar system with ultimate purpose to operate in real-time scenarios. Low-power and fully analog integrated architectures of the decision tree classification model were also introduced in order to classify the received data. The main building block of each classifier consists of a double-differential pair SFC, an analog MP and a WTA circuit. In order to evaluate the effectiveness of our approach, we conducted an extensive evaluation, pitting them against the established analog classifiers on the measured data. The model training and comparison against software-based classifier were executed within the Python programming environment. For the hardware design and subsequent post-layout simulation result processing, we made use of the Cadence IC Suite, employing a TSMC 90 nm CMOS process technology. Classification outputs demonstrate the effectiveness of the proposed architectures and validate the design methodology.

REFERENCES

[1] S. Abbate, M. Avvenuti, P. Corsini, J. Light, and A. Vecchio, "Monitoring of human movements for fall detection and activities recognition in elderly care using wireless sensor network: A survey," *Wireless Sensor Networks: Application-Centric Design*, vol. 1. London, U.K.: IntechOpen, 2010.

[2] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2688–2710, 2010.

[3] P. Kulurkar, C. kumar Dixit, V. Bharathi, A. Monikavishnuvarthini, A. Dhakne, and P. Preethi, "AI based elderly fall prediction system using wearable sensors: A smart home-care technology with IOT," *Meas., Sens.*, vol. 25, Feb. 2023, Art. no. 100614.

[4] B. Saha, M. S. Islam, A. K. Riad, S. Tahora, H. Shahriar, and S. Sneha, "BlockTheFall: Wearable device-based fall detection framework powered by machine learning and blockchain for elderly care," 2023, *arXiv:2306.06452*.

[5] R. C. King, E. Villeneuve, R. J. White, R. S. Sherratt, W. Holderbaum, and W. S. Harwin, "Application of data fusion techniques and technologies for wearable health monitoring," *Med. Eng. Phys.*, vol. 42, pp. 1–12, Apr. 2017.

[6] G. Wang, Q. Li, L. Wang, Y. Zhang, and Z. Liu, "Elderly fall detection with an accelerometer using lightweight neural networks," *Electronics*, vol. 8, no. 11, p. 1354, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/11/1354

[7] L. Palmerini, J. Klenk, C. Becker, and L. Chiari, "Accelerometer-based fall detection using machine learning: Training and testing on real-world falls," *Sensors*, vol. 20, no. 22, p. 6479, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/22/6479

[8] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and Accelerometer-derived posture information," in *Proc. 6th Int. Workshop Wearable Implantable Body Sensor Netw.*, 2009, pp. 138–143.

[9] B. Erol, M. Amin, F. Ahmad, and B. Boashash, "Radar fall detectors: A comparison," in *Proc. 20th SPIE Radar Sensor Technol.*, 2016, Art. no. 982918. [Online]. Available: https://doi.org/10.1117/12.2224984

[10] Y. Sun, R. Hang, Z. Li, M. Jin, and K. Xu, "Privacy-preserving fall detection with deep learning on mmWave radar signal," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, 2019, pp. 1–4.

[11] T. Imamura, V. G. Moshnyaga, and K. Hashimoto, "Automatic fall detection by using doppler-radar and LSTM-based recurrent neural network," in *Proc. IEEE 4th Glob. Conf. Life Sci. Technol. (LifeTech)*, 2022, pp. 36–37.

[12] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomed. Eng. Online*, vol. 12, no. 1, p. 66, 2013.

[13] W. Haensch, T. Gokmen, and R. Puri, "The next generation of deep learning hardware: Analog computing," *Proc. IEEE*, vol. 107, no. 1, pp. 108–122, 2018.

[14] B. J. MacLennan, "A review of analog computing," Dept. Electr. Eng. Comput. Sci., Univ. Tennessee, Knoxville, TN, USA, Rep. UT-CS-07-601, Sep. 2007.

[15] B. Ulmann, *Analog Computing*. Munich, Germany: Oldenbourg Wissenschaftsverlag, 2013.

[16] R. Sarpeshkar, "Analog synthetic biology," *Philosoph. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 372, no. 2012, 2014, Art. no. 20130110.

[17] T. York et al., "Bioinspired polarization imaging sensors: From circuits and optics to signal processing algorithms and biomedical applications," *Proc. IEEE*, vol. 102, no. 10, pp. 1450–1469, Oct. 2014.

[18] C. Prakash et al., "Computing of neuromorphic materials: An emerging approach for bioengineering solutions," *Mater. Adv.*, vol. 4, pp. 5882–5919, Oct. 2023.

[19] K. Li, Y. Pan, F. Chen, K.-T. Cheng, and R. Huan, "Real-time lossless ECG compression for low-power wearable medical devices based on adaptive region prediction," *Electron. Lett.*, vol. 50, no. 25, pp. 1904–1906, 2014.

[20] D. I. Fotiadis, C. Glaros, and A. Likas, "Wearable medical devices," *Wiley Encycl. Biomed. Eng.*, vol. 3, pp. 3816–3827, Apr. 2006.

[21] T. N. Gia et al., "Energy efficient wearable sensor node for IoT-based fall detection systems," *Microprocess. Microsyst.*, vol. 56, pp. 34–46, Feb. 2018.

[22] T. N. Gia et al., "IoT-based fall detection system with energy efficient sensor nodes," in *Proc. IEEE Nordic Circuits Syst. Conf. (NORCAS)*, 2016, pp. 1–6.

[23] S. Kamel Gharghan, S. Saad Fakhrulddin, A. Al-Naji, and J. Chahl, "Energy-efficient elderly fall detection system based on power reduction and wireless power transfer," *Sensors*, vol. 19, no. 20, p. 4452, 2019.

[24] D. G. Arnaoutoglou et al., "Acceleration-based low-cost CW radar system for real-time elderly fall detection," *IEEE J. Electromagn., RF Microw. Med. Biol.*, vol. 8, no. 2, pp. 102–112, Jun. 2024.

[25] P. Setlur, M. G. Amin, F. Ahmad, and P. D. Zemany, "Experiments on through-the-wall motion detection and ranging," in *Proc. 6th Sens., Command, Control, Commun., Intell. (C3I) Technol. Homel. Secur. Homel. Defense*, 2007, Art. no. 653809.

[26] H. Jeong, D. Kim, G. Kim, and S. Kim, "VitRad: A low-cost continuous wave doppler radar system with 3D-printed horn antennas for human vital sign detection," *HardwareX*, vol. 12, Oct. 2022, Art. no. e00361. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2468067222001067

[27] M. Skolnik, *Introduction to Radar Systems* (Electrical Engineering Series). New York, NY, USA: McGraw-Hill, 1980. [Online]. Available: https://books.google.gr/books?id=YvsiAAAAMAAJ

[28] B. Sadhu, M. Sturm, B. M. Sadler, and R. Harjani, "Analysis and design of a 5 GS/s analog charge-domain FFT for an SDR front-end in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 5, pp. 1199–1211, May 2013.

[29] M. Lehne and S. Raman, "A prototype analog/mixed-signal fast fourier transform processor IC for OFDM receivers," in *Proc. IEEE Radio Wireless Symp.*, 2008, pp. 803–806.

[30] P.G. Reshma, V. P. Gopi, V. S. Babu, and K. A. Wahid, "Analog CMOS implementation of FFT using cascode current mirror," *Microelectron. J.*, vol. 60, pp. 30–37, Feb. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0026269216307212

[31] B. Gilbert, "Translinear circuits: An historical overview," *Analog Integr. Circuits Signal Process.*, vol. 9, pp. 95–118, Mar. 1996.

[32] R. J. Wiegerink, *Analysis and Synthesis of MOS Translinear Circuits*, vol. 246. New York, NY, USA: Springer, 2012.

[33] A. Sarkar and S. Sengupta, "Second-degree digital differentiator-based power system frequency estimation under non-sinusoidal conditions," *IET Sci., Meas. Technol.*, vol. 4, no. 3, pp. 105–114, 2010.

[34] I. Nahlus, E. P. Kim, N. R. Shanbhag, and D. Blaauw, "Energy-efficient dot product computation using a switched analog circuit architecture," in *Proc. Int. Symp. Low Power Electron. Design*, 2014, pp. 315–318.

[35] Y. Zhang, N. Mirchandani, M. Onabajo, and A. Shrivastava, "RSSI amplifier design for a feature extraction technique to detect seizures with analog computing," in *Proc. IEEE Int. Symp. circuits Syst. (ISCAS)*, 2020, pp. 1–5.

[36] M. Yang, C.-H. Yeh, Y. Zhou, J. P. Cerqueira, A. A. Lazar, and M. Seok, "A 1μW voice activity detector using analog feature extraction and digital deep neural network," in *Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC)*, 2018, pp. 346–348.

[37] M. Yang et al., "Nanowatt acoustic inference sensing exploiting nonlinear analog feature extraction," *IEEE J. Solid-State Circuits*, vol. 56, no. 10, pp. 3123–3133, Oct. 2021.

[38] S. Ray and P. R. Kinget, "Ultra-low-power and compact-area analog audio feature extraction based on time-mode analog filterbank interpolation and time-mode analog rectification," *IEEE J. Solid-State Circuits*, vol. 58, no. 4, pp. 1025–1036, Apr. 2023.

[39] Z. Zeng, M. G. Amin, and T. Shan, "Arm motion classification using time-series analysis of the spectrogram frequency envelopes," *Remote Sens.*, vol. 12, no. 3, p. 454, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/3/454

[40] K. Hanifi and M. E. Karsligil, "Elderly fall detection with vital signs monitoring using CW doppler radar," *IEEE Sensors J.*, vol. 21, no. 15, pp. 16969–16978, Aug. 2021.

[41] D. Arnaoutoglou et al., "Toward real time processing of radar signals detecting elderly people fall," in *Proc. 18th IEEE Int. Workshop Cell. Nanoscale Netw. Appl. (CNNA)*, 2023, pp. 1–4.

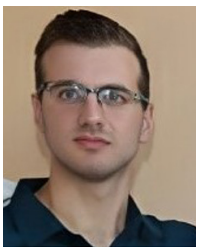
[42] Q. Wu, Y. D. Zhang, W. Tao, and M. G. Amin, "Radar-based fall detection based on doppler time–frequency signatures for assisted living," *IET Radar, Sonar Navig.*, vol. 9, no. 2, pp. 164–172, 2015. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-rsn.2014.0250

[43] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. New York, NY, USA: Springer, 2006.

[44] H. Sharma, S. Kumar et al., "A survey on decision tree algorithms of classification in data mining," *Int. J. Sci. Res.*, vol. 5, no. 4, pp. 2094–2097, 2016.

[45] S. Suthaharan, "Decision tree learning," in *Machine Learning Models and Algorithms for Big Data Classification*. Boston, MA, USA: Springer, 2016, pp. 237–269.
[46] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the Gini index and information gain criteria," *Ann. Math. Artif. Intell.*, vol. 41, pp. 77–93, May 2004.
[47] A. N. Soni, "Feature extraction methods for time series functions using machine learning," *Int. J. Innovat. Res. Sci., Eng. Technol.*, vol. 7, no. 8, pp. 8661–8665, 2018.
[48] I. Guyon and A. Elisseeff, "An introduction to feature extraction," in *Feature Extraction: Foundations and Applications*. Berlin, Germany: Springer, 2006, pp. 1–25.
[49] B. Panić, J. Klemenc, and M. Nagode, "Gaussian mixture model based classification revisited: Application to the bearing fault classification," *J. Mech. Eng./Strojniški Vestnik*, vol. 66, no. 4, pp. 215–226, 2020.
[50] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *Proc. Sci. Inf. Conf.*, 2014, pp. 372–378.
[51] E. Seevinck and R. J. Wiegerink, "Generalized translinear circuit principle," *IEEE J. Solid-State Circuits*, vol. 26, no. 8, pp. 1098–1102, Aug. 1991.
[52] V. Alimisis, G. Gennis, M. Gourdouparis, C. Dimas, and P. P. Sotiriadis, "A low-power analog integrated implementation of the support vector machine algorithm with on-chip learning tested on a bearing fault application," *Sensors*, vol. 23, no. 8, p. 3978, 2023.
[53] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. A. Mead, "Winner-take-all networks of O(*N*) complexity," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 1988, pp. 704–711.
[54] A. Wang, B. H. Calhoun, and A. P. Chandrakasan, *Sub-Threshold Design for Ultra Low-Power Systems*, vol. 95. New York, NY, USA: Springer, 2006.
[55] S.-C. Liu, J. Kramer, G. Indiveri, T. Delbrück, and R. Douglas, *Analog VLSI: Circuits and Principles*. Cambridge, MA, USA: MIT Press, 2002.
[56] V. Alimisis, C. Dimas, G. Pappas, and P. P. Sotiriadis, "Analog realization of fractional-order skin-electrode model for tetrapolar bio-impedance measurements," *Technologies*, vol. 8, no. 4, p. 61, 2020.
[57] I. Dimeas, I. Petras, and C. Psychalinos, "New analog implementation technique for fractional-order controller: A DC motor control," *AEU-Int. J. Electron. Commun.*, vol. 78, pp. 192–200, 2017.
[58] M. Hock, A. Hartel, J. Schemmel, and K. Meier, "An analog dynamic memory array for neuromorphic hardware," in *Proc. Eur. Conf. Circuit Theory Design (ECCTD)*, 2013, pp. 1–4.
[59] G. Indiveri, "A current-mode hysteretic winner-take-all network, with excitatory and inhibitory coupling," *Analog Integr. Circuits Signal Process.*, vol. 28, no. 3, pp. 279–291, 2001.
[60] M. Akbari, T.-I. Chou, and K.-T. Tang, "An adjustable 0.3 V current winner-take-all circuit for analogue neural networks," *Electron. Lett.*, vol. 57, no. 18, pp. 685–687, 2021.
[61] A. Tajalli and Y. Leblebici, *Extreme Low-Power Mixed Signal IC Design: Subthreshold Source-Coupled Circuits*. New York, NY, USA: Springer, 2010.
[62] C. Mead, "Analog VLSI and neutral systems," *NASA STI/Recon Tech. Rep. A*, vol. 90, Jan. 1989, Art. no. 16574.
[63] K. Kang and T. Shibata, "An on-chip-trainable Gaussian-Kernel analog support vector machine," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1513–1524, Jul. 2010.
[64] V. Alimisis, G. Gennis, C. Dimas, and P. P. Sotiriadis, "An analog Bayesian classifier implementation, for thyroid disease detection, based on a low-power, current-mode gaussian function circuit," in *Proc. Int. Conf. Microelectron. (ICM)*, 2021, pp. 153–156.
[65] V. Alimisis, G. Gennis, K. Touloupas, C. Dimas, M. Gourdouparis, and P. P. Sotiriadis, "Gaussian mixture model classifier analog integrated low-power implementation with applications in fault management detection," *Microelectron. J.*, vol. 126, Aug. 2022, Art. no. 105510.
[66] S.-Y. Peng, P. E. Hasler, and D. V. Anderson, "An analog programmable multidimensional radial basis function based classifier," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 10, pp. 2148–2158, 2007.
[67] A. Reda, L. Qi, Y. Li, and G. Wang, "A generic Nano-watt power fully tunable 1-D Gaussian kernel circuit for artificial neural network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 9, pp. 1529–1533, Sep. 2020.
[68] K. Lee, J. Park, and H.-J. Yoo, "A low-power, mixed-mode neural network classifier for robust scene classification," *J. Semicond. Technol. Sci.*, vol. 19, no. 1, pp. 129–136, 2019.
[69] R. Zhang and T. Shibata, "An analog on-line-learning K-means processor employing fully parallel self-converging circuitry," *Analog Integr. Circuits Signal Process.*, vol. 75, pp. 267–277, May 2013.
[70] R. Zhang, N. Uetake, T. Nakada, and Y. Nakashima, "Design of programmable analog calculation unit by implementing support vector regression for approximate computing," *IEEE Micro*, vol. 38, no. 6, pp. 73–82, Nov./Dec. 2018.
[71] F. Li, C.-H. Chang, and L. Siek, "A compact current mode neuron circuit with gaussian taper learning capability," in *Proc. IEEE Int. Symp. circuits Syst.*, 2009, pp. 2129–2132.
[72] Z. Zhao, A. Srivastava, L. Peng, and Q. Chen, "Long short-term memory network design for analog computing," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 15, no. 1, pp. 1–27, 2019.
[73] E. Georgakilas, V. Alimisis, G. Gennis, C. Aletraris, C. Dimas, and P. P. Sotiriadis, "An ultra-low power fully-programmable analog general purpose type-2 fuzzy inference system," *AEU-Int. J. Electron. Commun.*, vol. 170, Oct. 2023, Art. no. 154824.
[74] V. Alimisis, G. Gennis, E. Tsouvalas, C. Dimas, and P. P. Sotiriadis, "An analog, low-power threshold classifier tested on a bank note authentication dataset," in *Proc. Int. Conf. Microelectron. (ICM)*, 2022, pp. 66–69.
[75] V. Alimisis, V. Mouzakis, G. Gennis, E. Tsouvalas, and P. P. Sotiriadis, "An analog nearest class with multiple Centroids classifier implementation, for depth of anesthesia monitoring," in *Proc. Int. Conf. Smart Syst. Power Manage. (IC2SPM)*, 2022, pp. 176–181.
[76] V. Alimisis, M. Gourdouparis, G. Gennis, C. Dimas, and P. P. Sotiriadis, "Analog Gaussian function circuit: Architectures, operating principles and applications," *Electronics*, vol. 10, no. 20, p. 2530, 2021.
[77] T. Davies and T. Fearn, "Back to basics: The principles of principal component analysis," *Spectrosc. Eur.*, vol. 16, no. 6, p. 20, 2004.

VASSILIS ALIMISIS (Student Member, IEEE) received the B.Sc. degree in physics (top 1%) and the M.Sc. degree in electronics and communications from the University of Patras, Greece, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with the National Technical University of Athens (NTUA), Greece, under the supervision of Prof. P. P. Sotiriadis. His Ph.D. thesis and research are supported and financed by the E.L.K.E. NTUA Scholarships. He is a Teaching Assistant in undergraduate and graduate courses and supervises Diploma Thesis. He has authored and coauthored several conference papers and journal articles. His main research interests include analog microelectronic circuits, low-power electronics, biomedical circuits and systems, analog computing, and integrated circuit architectures with applications in artificial intelligence and machine learning. He has received the Best Paper Award at the IEEE International Conference on Microelectronics 2020, the Best Paper Award at the IEEE International Conference on Microelectronics 2021, the Best Paper Award (3rd Place) at the IEEE International Conference on Microelectronics 2023, the Best Paper Award at the IEEE Symposium on Integrated Circuits and Systems Design (SBCCI) 2021, and the Best Paper Award at the 1st International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications in 2023. He regularly reviews for many IEEE TRANSACTIONS and conferences and serves on proposal review panels.



DIMITRIOS G. ARNAOUTOGLOU (Graduate Student Member, IEEE) was born in Thessaloniki, Greece, in September 1998. He received the B.S. degree from the Democritus University of Thrace, Xanthi, Greece, in 2021, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include multiferroics devices, microwave systems for space applications, and biomedical microwaves applications.

EMMANOUIL ANASTASIOS SERLIS (Student Member, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 2023, where he is currently pursuing the Ph.D. degree under the supervision of Prof. P. P. Sotiriadis. He has coauthored several conference papers and journal articles. His main research interests include analog microelectronic circuits, ultralow-power electronics, analog computing, and integrated circuit architectures with applications in artificial intelligence, machine learning, and deep neural networks.

ARGYRO KAMPERI (Student Member, IEEE) is currently pursuing the Diploma degree with the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece, under the supervision of Prof. P. P. Sotiriadis. She has coauthored several conference papers and journal articles. Her main research interests include analog microelectronic circuits, ultralow-power electronics, analog computing, and low-power analog integrated filters. She has received the Best Paper Award at the 1st International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications in 2023.

KONSTANTINOS METAXAS (Student Member, IEEE) is currently pursuing the Diploma degree with the Department of Electrical and Computer Engineering, National Technical University of Athens, Greece, under the supervision of Prof. P. P. Sotiriadis. He has coauthored several conference papers and journal articles. His main research interests include analog microelectronic circuits, ultralow-power electronics, analog computing, nonlinear dynamical systems, and synchronization. He has received awards and honors for his academic achievements during his studies.

GEORGE A. KYRIACOU (Senior Member, IEEE) was born in Famagusta, Cyprus, in March 1959. He received the Electrical Engineering Diploma and Ph.D. degrees (Hons.) from the Democritus University of Thrace, Xanthi, Greece, in 1984 and 1988, respectively.

Since January 1990, he has been with the Department of Electrical and Computer Engineering, Democritus University of Thrace, where he is currently a Professor and the Director of the Microwaves Laboratory and was the Director of the Graduate Studies from 2005 to 2010. He has authored more than 250 journal articles and conference papers and supervised 8 Ph.D. and 15 M.Sc. theses and more than 110 Diploma theses. His main research interests include microwave engineering, open waveguides and antennas in anisotropic media, software defined and cognitive radio, computational electromagnetics, and biomedical engineering. He has been serving as an Associate Editor for *IET Microwaves, Antennas and Propagation* since 2015. He is a member of the Technical Chamber of Greece and the European Microwave Association.

PAUL P. SOTIRIADIS (Fellow, IEEE) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens (NTUA), Greece, in 1994, the M.S. degree in electrical engineering from Stanford University, USA, in 1996, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, USA, in 2002.

He is a Professor of Electrical and Computer Engineering with NTUA, the Director of the Electronics Laboratory, NTUA, and a Governing Board Member of the Hellenic (National) Space Center of Greece. He runs a team of 25 researchers. In 2002, he joined the faculty of the Electrical and Computer Engineering Department, Johns Hopkins University and in 2012, he joined the faculty of the Electrical and Computer Engineering Department, NTUA. He has authored and coauthored more than 200 research publications, most of them in IEEE journals and conferences, holds one patent, and has contributed several chapters to technical books. His research interests include the design, optimization, and mathematical modeling of analog, mixed-signal, and RF integrated and discrete circuits, sensor and instrumentation architectures with emphasis in biomedical instrumentation, advanced RF frequency synthesis, and, the application of machine learning and general AI in the operation as well as the design of electronic circuits.

Prof. Sotiriadis has received several awards, including the prestigious Guillemin-Cauer Award from the IEEE Circuits and Systems Society in 2012, the Best Paper Award at the 1st International Conference on Frontiers of Artificial Intelligence, Ethics, and Multidisciplinary Applications in 2023, the Best Paper Award (3rd Place) at the IEEE International Conference on Microelectronics 2023, the Best Paper Award at the IEEE International Conference on Microelectronics 2021, the Best Paper Award at the IEEE Symposium on Integrated Circuits and Systems Design (SBCCI) 2021, the Best Paper Award at the IEEE International Conference on Microelectronics 2020, the Best Paper Award at the IEEE International Conference on Modern Circuits and Systems Technologies 2019, the Best Paper Award at the IEEE International Frequency Control Symposium 2012, the Best Paper Award at the IEEE International Symposium on Circuits and Systems 2007, and the IEEE Circuits and Systems Society Outstanding Technical Committee Recognition 2022. He has also been in the list of the top 2% most influential researchers in the world in 2020, 2022, and 2023. He is an Associate Editor of the IEEE SENSORS JOURNAL and IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS, has served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART I: REGULAR PAPERS from 2016 to 2020 and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS from 2005 to 2010, and has been a member of technical committees of many conferences. He regularly reviews for many IEEE TRANSACTIONS and conferences and serves on proposal review panels.