Machine Learning on FPGA for Robust Si_3N_4-Gate ISFET pH Sensor in Industrial IoT Applications

Soumendu Sinha, *Member, IEEE*, Nishad Sahu, Rishabh Bhardwaj, Aditya Mehta, Hitesh Ahuja, Satyam Srivastava, Anubhav Elhence, and Vinay Chamola, *Senior Member, IEEE*

***Abstract*—This article presents performance enhancement of Si_3N_4-gate ion-sensitive field-effect transistor based pH sensor using machine learning (ML) techniques. A robust SPICE macromodel is developed using experimental data, which incorporates intrinsic temperature and temporal characteristics of the device, which is further used in sensor readout circuit (ROIC), which shows a nonideal temperature and time dependence in the voltage output. To make the device robust to the critical drifts, we exploit six state-of-the-art ML models, which are trained using the data generated from ROIC for a wide range of pH, temperature, and temporal conditions. Thorough comparison between ML models shows random forest outperforms other models for drift compensation task. This work also shows a preliminary time series classification task. The ML models are implemented on a Xilinx PYNQ-Z1 field-programmable gate array (FPGA) board to validate the performance in power and memory-restricted environment, crucial for IoT applications. A parameter, implementation factor is defined to evaluate best ML model for IoT deployment using FPGA/MCU hardware implementation. The significantly lower power consumption of FPGA board as compared to CPU with no noticeable performance drop is a pointer to the future of robust pH sensors used in industrial and remote IoT applications.**

***Index Terms*—Drift compensation, field-programmable gate array (FPGA), ion-sensitive field-effect transistor (ISFET), IIoT, IoT, machine learning (ML), MCU, neural networks, pH sensor, random forest (RF), SPICE macromodel.**

Manuscript received May 19, 2021; revised August 15, 2021; accepted September 27, 2021. Date of publication October 1, 2021; date of current version November 19, 2021. This work was supported in part by the Council of Scientific and Industrial Research, New Delhi, India. Paper 2021-IACC-0430.R1, presented at the 2020 IEEE 17th India Council International Conference, New Delhi, India, Dec. 11–13, and approved for publication in the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS by the Industrial Automation and Control Committee of the IEEE Industry Applications Society. *(Corresponding author: Soumendu Sinha.)*

Soumendu Sinha is with the Semiconductor Devices Area, CSIR-Central Electronics Engineering Research Institute, Pilani 333031, India (e-mail: soumendu@ceeri.res.in).

Nishad Sahu, Hitesh Ahuja, and Anubhav Elhence are with the Department of Electrical and Electronics, Birla Institute of Technology and Science, Pilani, Pilani 333031, India (e-mail: h20160215@pilani.bits-pilani.ac.in; h20190070@pilani.bits-pilani.ac.in; f2015499@pilani.bits-pilani.ac.in).

Rishabh Bhardwaj is with the Information Systems Technology and Design, Singapore University of Technology and Design, Singapore 487372, Singapore (e-mail: rishabh_bhardwaj@mymail.sutd.edu.sg).

Aditya Mehta is with the Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani, Pilani 333031, India (e-mail: f20170783@pilani.bits-pilani.ac.in).

Satyam Srivastava is with the Intelligent Systems Group, CSIR-Central Electronics Engineering Research Institute, Pilani 333031, India (e-mail: satyamsrivastava@ceeri.res.in).

Vinay Chamola is with the Department of Electrical and Electronics Engineering and the Anuradha and Prashanth Palakurthi Centre for Artificial Intelligence Research, Birla Institute of Technology and Science, Pilani, Pilani 333031, India (e-mail: vinay.chamola@pilani.bits-pilani.ac.in).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIA.2021.3117233.

Digital Object Identifier 10.1109/TIA.2021.3117233

I. Introduction

THE field of biomedical and chemical sensing applications has been a fascinating area for research since several decades [2]. With the advancement in micro/nanofabrication technologies, it requires that miniaturized sensors should not compromise on the measurement accuracy and precision [3]. Electrochemical sensors are widely used for pH sensing applications [4]. Ion-sensitive field-effect transistor (ISFET) is one of the most promising electrochemical sensors due to inherent advantages of easy fabrication, low cost, and small size [5], [6] for chemical/biochemical sensing applications, popularly utilized for pH sensing applications, as listed in Table I.

Despite the wide range of applications, ISFETs are prone to temperature and temporal drift, restricting their accuracy in field applications [7]. ISFETs show nonlinear drift characteristics due to temperature-dependent semiconductor and electrochemical parameters [8], [9]. The temporal drift in ISFET devices is due to chemical modification of the sensing film [10]. In order to deploy ISFET-based pH sensors for critical field applications, such as biomedical devices, environmental monitoring, and food quality monitoring, robust drift compensation methodologies are critical to provide accurate sensor data [5], [11]. Manual calibration of such deviations is difficult when remotely deployed in large numbers, and the nonlinear drift behavior makes it extremely challenging with varying temperature and time in field deployment [8]. Therefore, artificial intelligence assisted techniques are increasingly being utilized for automated sensor calibration [12].

Machine learning (ML) techniques are powering major transformations in the novel era of automation. The algorithms have proven to be extremely effective in learning impact of independent variables on the dependent variables [13]. As ISFET-readout circuit (ROIC) shows output voltage (pH) dependence on ambient conditions, we infer that ML techniques greatly suits the compensation task [7]. From the data generated using device-ROIC, ML algorithm learns approximate input–output functions in the training phase. To test a model's performance, i.e., testing phase, we evaluate the model on an unseen test data. Previous works have seen importance of ML-based approaches in temperature and temporal drift compensation [14].

0093-9994 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

TABLE I
TABLE LISTING OUT THE APPLICATIONS OF ISFET IN VARIOUS INDUSTRIES

#	Industrial Application	Application Description	Ref.
1.	**Bio-medical / clinical, diagnosis and research**	Rapid detection of SARS-CoV-2 virus causing the COVID-19 disease	[15–17]
		Malaria diagnosis	[18]
		Human Genome Project and genome/DNA sequencing	[6, 19–22]
		Accuracte and high throughput pH sensning	[19]
		Robust label-free Micro RNA detection	[23]
		Breast cancer mutation detection	[24]
		pH and oxygen partial pressure measurement	[25]
		Point of care biomedical applications	[26]
		DNA methylation detection	[27]
2.	**Agriculture**	Development of wireless sensor nodes for algae cultivation	[28]
		Precision Agriculture	[29–31]
		Soil and Crops Measuements	[32]
		In situ monitoring of soil nutrients/ soil nitrate sensing/ soil analysis/soil sensor	[33–37]
3.	**Environmental monitoring**	Monitoring of wastewater	[32]
		Geochemical Barriers Monitoring	[32]
		Pesticide and toxin concentration detection	[38]
		Coastal pH monitoring	[39]
		Pollution monitoring in liquid media	[40]
		Sea water alkalinity measurement	[41]
4.	**Wearable sensing**	Sweat sensing	[42, 43]
		Skin temperature monitoring	[43]
		On body pH measurement	[44]
		Wearable Point of care applications	[45]
		Body Fluid pH monitoring	[46]
5.	**Food**	Heavy metal ions detection in vegetables	[47]
		Determination of glucose, ascorbic and citric acids	[48]
		Controlling Phosphororgannic pesticides in Water and Vegetables	[49]
		Nitrate ion determination in vegetables	[50]
6.	**Horticulture**	pH and potassium measurement in nutrient solution	[51]
		Multi ion sensing	[52]
		Soil Analysis	[53]
		Nutrient sensing for rockwool culture	[54]
		Application in closed loop system for green houses	[55]

For the compensation task, we compare the accuracy of various models, viz., multilayer perceptron (MLP), linear regression (LR), polynomial regression, decision trees (DTs), random forest (RF), and support vector machine (SVM). For the purpose of data generation, a robust SPICE macromodel of Si_3N_4-gate ISFET has been developed using experimental data, which includes the temperature and temporal dependence of electrochemical and device parameters [8]. The model is essential to generate quality data for later used ML algorithms for the task of concern. This SPICE macromodel of ISFET is exported as a subcircuit block in a constant voltage constant current (CVCC) ROIC. In comparison to our previously reported work [1], we extend this work to the implementation and performance comparison of the ML models on field-programmable gate array (FPGA)/MCUs using Xilinx PYNQ-Z1 board for drift compensation in sensors and study feasibility for IoT deployment.

To the best of our knowledge, our five major contributions are as follows.

1) We introduce ML-based models to significantly reduce temperature and temporal dependence of Si_3N_4-gate ISFET pH sensor in a CVCC ROIC.

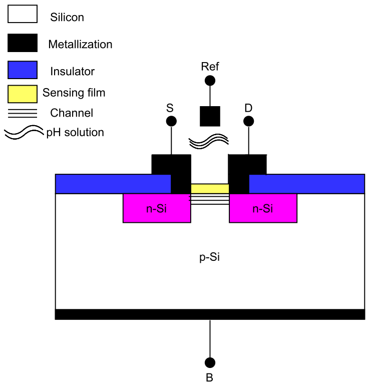


Fig. 1. Schematic of the cross-sectional view of ISFET device [1].

2) Based on our compensated results, we compare the performance of various ML models to evaluate the best performing ML technique for Si_3N_4-gate ISFET-pH sensor.
3) As a preliminary work, we show how by only using drifted values of the reference voltage, we can estimate the pH of the solution.
4) We also present for the first time implementation of ML models on FPGA using Xilinx PYNQ-Z1 board for drift compensation in sensors and compare the results in terms of accuracy, speed, and power consumption with CPU, MCU, and FPGA implementation.
5) We also propose a novel factor called implementation factor (IF) to evaluate the performance of ML models for FPGA/MCU implementation. The factor is calculated by giving equal weights to accuracy, speed, and energy consumption of the ML models on the FPGA.

This article organization is as follows. Section II presents an overview of ISFET devices and its SPICE macromodel formulation along with the details of ROIC topology. Section III presents an overview of the ML algorithms and the experimental setups for drift compensation tasks. Section IV discusses the FPGA and MCU implementation of the ML models. Section V presents the results obtained from the various experimental setups discussed. Finally, Section VI concludes this article.

II. ISFET DEVICE DESCRIPTION AND MACROMODEL FORMULATION

ISFET device is similar to a MOSFET in terms of the device structure, except the gate electrode in the MOSFET is replaced by a reference electrode, a sensing layer, and an ionic solution, as shown in Fig. 1 [8]. A reference electrode is inserted in the ionic solution to supply a reference potential. The ionic solution is in contact with the gate oxide, which leads to the formation of charges at the solution/oxide interface, which further modulates the threshold voltage of the ISFET, causing variation in the channel current, which is sensed by a suitable readout circuit, by sensing the voltage at the output terminal.

The semiconductor and the electrochemical parameters of ISFET have a dependence on temperature [9]. Thus, the threshold voltage of ISFET gets affected by the concentration of ions

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

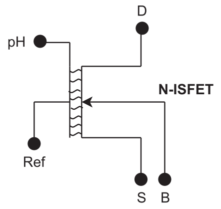


Fig. 2. Schematic of the developed ISFET SPICE Macromodel [1].

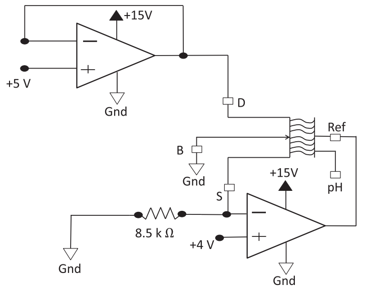


Fig. 3. CVCC topology for readout of ISFET sensor [1].

in the electrolyte solution and the temperature [10]. Sensing films, such as Si_3N_4, also show a significant temporal drift [8] due to inherent transient nature of chemical reactions taking place at the surface of the sensing film, leading to chemical modification of insulator with time, which affects the device threshold voltage [10].

A. SPICE Macromodel Formulation of Si_3N_4-Gate ISFET

The macromodel of ISFET is developed in SPICE because it gives us the flexibility to define custom subcircuit blocks, which contains robust modeling of all the parameters pertaining to the circuit elements involved. Sinha *et al.* carried out comprehensive modeling of the dependence of the parameters present in the both electrochemical and electronic stages, considering the parameters' dependence on temperature and time in [8] to develop an accurate Si_3N_4-gate ISFET SPICE macromodel that closely mimics real-world ISFET device. The study uses experimental data to develop a robust SPICE macromodel and the simulated characteristics of the SPICE model is very close to the experimental results. We follow a similar approach and use the developed SPICE macromodel in this study. The ISFET SPICE model consists of five nodes, namely drain (D), source (S), bulk (B), the reference electrode potential connected to the electronic stage of the device via electrical double-layer capacitors, and pH, which is the linear mapping of pH of the solution as an input voltage terminal. In CVCC circuit topology, the subcircuit block shown in Fig. 2 is used to obtain the sensor response [7].

B. ISFET Readout Circuit Topology

CVCC topology (see Fig. 3) is used as a readout circuit derived from previous study [8]. The voltages at the source and the drain nodes of the device are kept constant using op-amps in negative feedback and voltage buffer modes, respectively. Hence, due to constant V_{ds} and I_{ds}, the quantity $(V_{\mathrm{ref}} - V_{\mathrm{t}})$ is fixed. Thus, any change in the threshold voltage due to varying temperature, time, and pH is reflected by change in V_{ref} of the device. The V_{ref} output relates to the pH of the solution when the temperature and time are constant. For the same pH solution, when the temperature and time are varied, the V_{ref} values obtained correspond to erroneous values of measured pH, which needs to be corrected through intelligent techniques. The ROIC output is given as an input to the ML algorithms to compensate the undesired temporal and temperature drift.

C. Accuracy of Si_3N_4-Gate ISFET SPICE Macromodel

The SPICE macromodel used in this study is based on our previous study on development of an accurate Si_3N_4-gate pH-ISFET device [8]. The results of Sinha *et al.* [8] indicate that the model is able to obtain characteristics very close to the experimental data for both temperature and temporal drift. For temperature drift, the model was developed based on the experimental data reported in [56]. In the transfer characteristics of the device, the simulated isothermal point was obtained at (4.25 V, 457.45 μA) against the experimentally reported value of (4.10 V, 468.26 μA), i.e., an error of 3.53% in the gate voltage and error of 2.29% in the drain to source current, which is within the tolerance limits.

For the temporal drift, the experimental data were taken from Jamasb *et al.* [57]. Using the experimental data, the parameters were extracted for modeling the temporal drift in the ISFET SPICE macromodel, which provided an Adj. R^2 value of close to 1. Thus, the simulated temporal drift was very close to the experimental data, which is confirmed with a high value of Adj. R^2.

Thus, the SPICE macromodel reported in [8] gives very close accuracy to the real ISFET device characteristics. An in-depth explanation of the developed mathematical model used in the ISFET SPICE macromodel for the calculation of ISFET threshold voltage, temporal drift, and temperature drift is provided in [7] and [8]. The developed model has been used in this study for generating the data for training the ML models. Thus, a successful drift compensation on this dataset can be emulated with high confidence for drift compensation in real-life ISFET-pH sensors.

III. Experimental Setup

We formulate the compensation task as two subtasks. The first task focuses on compensating the voltage drift induced by only considering one parameter, i.e., temperature, whereas the second task aims to tackle a more complicated challenge of offsetting voltage drift caused by variability in both temperature and time. Since both time and temperature lead to voltage drift, it is a nonlinear function estimation problem, thereby increasing its complexity.

All models were built using the Sklearn Python library in Python 3.5 [58]. In this study, we compare task performances of six state-of-the-art ML models described in this section. The pipeline of approach is depicted in Fig. 4. We encourage reader

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

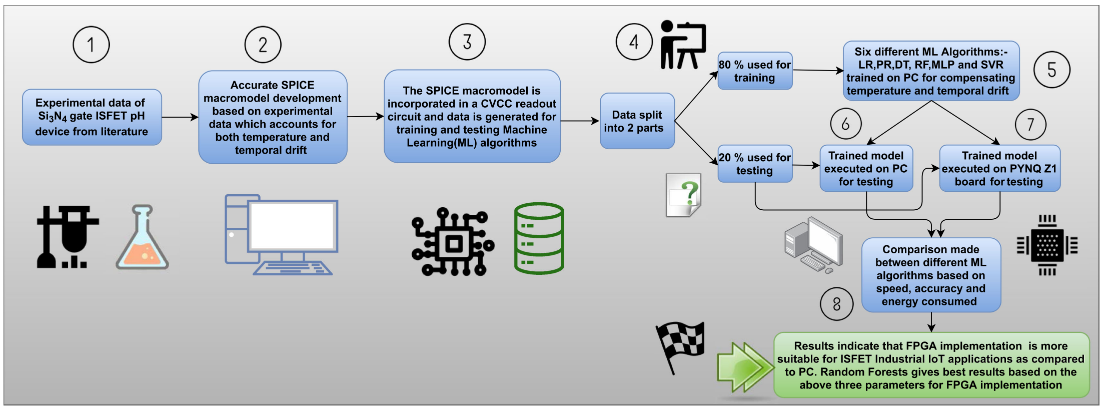

1
Experimental data of Si_3N_4 gate ISFET pH device from literature
2
Accurate SPICE macromodel development based on experimental data which accounts for both temperature and temporal drift
3
The SPICE macromodel is incorporated in a CVCC readout circuit and data is generated for training and testing Machine Learning(ML) algorithms
4
Data split into 2 parts
80 % used for training
20 % used for testing
Six different ML Algorithms:- LR,PR,DT, RF,MLP and SVR trained on PC for compensating temperature and temporal drift
5
6
Trained model executed on PC for testing
7
Trained model executed on PYNQ Z1 board for testing
Comparison made between different ML algorithms based on speed, accuracy and energy consumed
8
Results indicate that FPGA implementation is more suitable for ISFET Industrial IoT applications as compared to PC. Random Forests gives best results based on the above three parameters for FPGA implementation

Fig. 4. Pipeline of the adopted methodology.

to refer the work in [58] for fine-grained details about ML model and function optimization that lead to model parameter learning.

A. *ML Algorithms*

1) Linear (LR) and Polynomial Regression (PR)*:* LR is the most basic form of regression technique, which involves fitting a straight line on the training data with a strong assumption, i.e., linear relationship between input–output variables. PR is an extension to LR incorporating nonlinear dependencies. The algorithm tries to fit a polynomial curve on the data provided by optimizing a cost function. If $x = \{x_1, \ldots, x_n\}$ is the input vector and y be the output

$$y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2^2 + \cdots + \alpha_n x_n^2 \tag{1}$$

where α_i are weights that are learnt during training the regressor model.

2) Support Vector Regression (SVR): The algorithm works on a similar principle as SVM. In SVM, the algorithm finds a hyperplane, with function $f(x) = wx + b$, given $f(x)$ lies within margin of tolerance (ϵ) from the value $y(x)$ of every data point, with the greatest possible margin between the decision line and the nearest point from it from samples representing a specific class. The hyperplane with the maximum distance margin will classify the data points correctly. However, in the case of SVR, the support vectors come up with the closest match between the data points and the actual function that is represented by them. So, maximizing the distance between the support vectors to the regressed curve, we move toward the actual curve. Mathematically, this can be represented as

$$\text{minimize } f = \frac{1}{2}\|w\|^2 \tag{2}$$

$$\text{s.t. } y_i - w_1 \cdot x_i - b \leq \varepsilon;\ w_1 \cdot x_i + b - y_i \leq \varepsilon. \tag{3}$$

3) Decision Trees: This algorithm builds the models in the form of an inverted tree structure. Each branch of the tree represents a possible decision, and with such branches, the associated DT is incrementally developed. The leaf nodes at the bottom represent the final decision. In regression problem, the criteria to split the branches are usually decided by selecting the setup resulting in minimum mean square error (MSE) or standard deviation.

4) Random Forest: RF constructs multiple DTs during training phase. It fits these DT on subsamples of data and gives the averaged output for the computation of accuracy. There are two key concepts that cause the randomness. First, there is random sampling of training data at the time of building trees. Second, there are random subsets of features examined when splitting nodes. For an input, select with replacement K random samples from the training set followed by learning a separate DT. Prediction on an unseen sample x_{new} can be made by averaging prediction from all the individual DT on x_{new}

$$\hat{f}(x_{\text{new}}) = \frac{1}{K} \sum_{k \in \{1,\ldots K\}} f_k(x_{\text{new}}). \tag{4}$$

5) Multilayer Perceptron: An MLP is a feed-forward artificial neural network model that maps the input data to the output data. The framework of MLP consists of neurons, which are the basic units of computations. This fundamental unit receives input from other neurons or an external source and computes the output. Each of the inputs have an associated weight, which is being imposed on the bias of its comparative importance to other inputs. A neuron performs the following operation:

$$f(x) = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{5}$$

where w represents the weight vector, x represents the input vector, and b adds the bias φ is the activation function.

6) Recurrent Neural Networks (RNNs): Standard feed-forward neural networks fail to capture sequential structure of the input. To tackle problems where the input has a series of values having notion of sequence, a special class of neural network was designed. An RNN (directionally) iterates over the element of the sequence and encodes the necessary information from the sequence. The input is processed as follows:

$$h_t = f_W(h_{t-1}, v_t) \tag{6}$$

$$V_p = f_O(h_n) \tag{7}$$

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

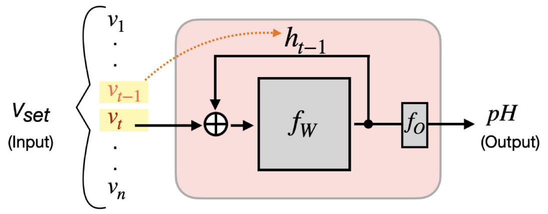


Fig. 5. Schematic of RNN architecture.

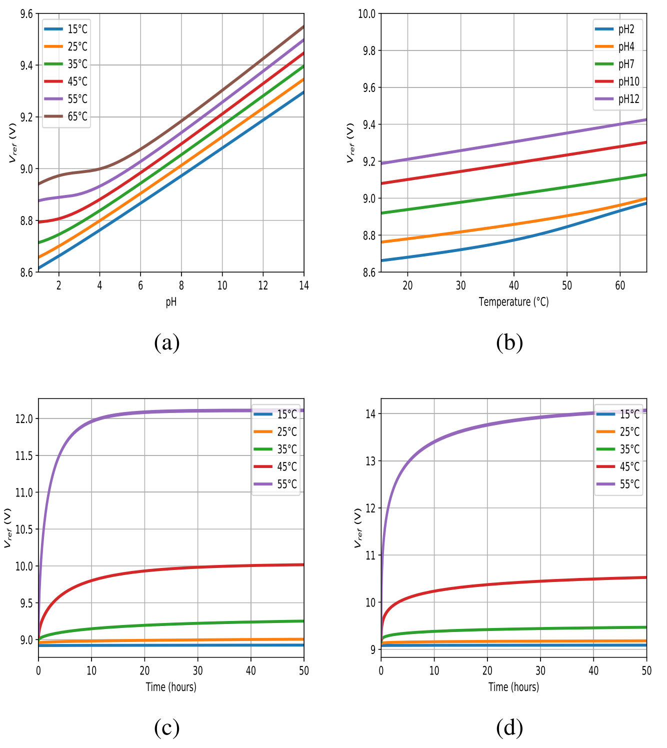


Fig. 6. Characteristic curves of $\mathrm{Si_3N_4}$-based ISFET. (a) V_{ref}-pH curves at different temperatures. (b) V_{ref}-temperature curves at different pH. (c) V_{ref}-time curves at different temperatures for pH 7. (d) V_{ref}-time curves at different temperatures for pH 10 [1].

where f_W and f_O are the function with trainable parameters W and O, respectively. f_O learns to map hidden representation of input. h_t is the hidden RNN generates until sequence at step t, i.e., v_t. For prediction, we use h_{n_t} and n_t being the sequence length. We feed h_{n_t} to a feed-forward NN with softmax activation to predict the pH class (see Fig. 5).

We used mean absolute error (MAE), MSE, and R^2 metrics to test the aforementioned models [59].

B. Experiment 1—Temperature Drift Compensation

As discussed earlier, temperature is one of the primary factors leading to ROIC output drift provided by an ISFET sensor. We plot the temperature-dependent characteristics of ISFET ROIC, as shown in Fig. 6(a) and (b). We tested several ML models in this first experiment, which attempts to predict the actual pH by compensating for V_{ref} drift. Here, we have used the data points from the set of reference voltages and corresponding pH of the solution, ranging from 0 to 14 at a resolution of 0.01. The

TABLE II
SUMMARIZING EXPERIMENTS FOR INPUT AND OUTPUT FEATURES

Experiment	Input features	Output
1:Temp Comp	Temperature+V_{ref}	pH
2:Temp-time Comp	Temperature+V_{ref}+Time	pH-shift
3:Series classification	Sequence of V_{ref} values	pH

temperature is varied from 15 to 65°C. Hence, the total number of samples obtained were 66 351. The whole data were split into training and testing datasets in the ratio of 80:20%. For efficient preparation, preprocessing techniques, such as normalization and mean centering, were taken into account. We have taken ROIC output voltage and ambient temperature as input features for the algorithms.

C. Experiment 2—Temperature–Temporal Drift Compensation

For this experiment, we have taken data points for temperature ranging from 15 to 65°C and time varying from 0 to 50 h. We plot temperature–temporal dependent ROIC output, as shown in Fig. 6(c) and (d). Data corresponding to pH 7 and 10 were considered for this experiment, which resulted in a total of 81 254 data samples. Again, the data are split in the ratio 80:20% for training and testing purposes and similar preprocessing as in Experiment 1 was applied. We have considered pH, ambient temperature, and time as the input features for training all the models, with an aim to capture the changing voltage for a particular pH with change in time and temperature.

D. Experiment 3—Time Series Classification

Owing to the limited temporal data for pH values 7 and 10, we also performed time-series classification. Given a set of ten consecutive V_{ref} reading given by the pH sensor, the binary classification task is to predict the pH of the solution. The problem formulation is slightly harder from the previous two experiments as we do not have the time stamp information at which the readings are noted, and also the data are scarce. For the time-series (sequence) classification, we use RNNs. The input consists of two feature vectors, one corresponding to the ten consecutive V_{ref} readings and the other is the natural logarithm of each voltage reading. As we suspect the voltage should follow an exponential decay curve, the logarithm of the voltage might show linear dependence with time. This comes from the idea that learning linear relationships is significantly easier (since it facilitates analytical solutions) than capturing a nonlinear relationship between dependent and independent variables from the dataset. We sample 2500 nonoverlapping sequences of V_{ref} through time and for both the pH 7 and 10. Training set constitutes randomly shuffled 70% of the dataset samples, 10% is used for development, and 20% for performance testing.

We have summarized the input features and expected output of the aforementioned experiments in Table III.

IV. FPGA AND MCU IMPLEMENTATION OF ML ALGORITHMS FOR IOT DEPLOYMENT

In addition to conventional implementation on CPU, we also implemented the algorithms discussed in Section III on FPGA

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PL RESOURCE IN THE PYNQ Z1 BOARD [60], [61]

Sl. No.	Programmable logic resource	No. of Units
1.	6-input LUTs	53,200
2.	Flip-flops	106,400
3.	BRAM (block RAM)	630 KB
4.	Clock management tiles each with a phase-locked loop(PLL) and mixed-mode-clock manager (MMCM)	4
5.	DSP slices	220
6.	On chip analog-to digital converter	1

and MCU hardware. The ML models discussed in this study are relatively simple and, therefore, in addition to FPGAs, they can also be implemented using microcontrollers with specialized instruction set, although it may take long time. In this study, we choose to implement them on programmable system (PS) and programmable logic (PL) of the Xilinx PYNQ-Z1 board. The implementation on the PS is considered as an MCU implementation, whereas the implementation on the PS+PL is considered as an FPGA implementation. FPGAs provide the feature of rapid prototyping. In comparison to FPGAs, microcontrollers with specialize instruction set consume lesser area but consume more time as well as power for implementing the same ML algorithms. PYNQ (Python productivity for ZYNQ) is the latest open-source project from Xilinx, which makes it easier to program FPGAs using the Python language [60]. Traditionally, Verilog HDL code was essential to be written for implementing algorithms on hardware, such as FPGAs, but with the aid of the PYNQ framework, such algorithms can now be directly programmed into FPGA using languages, such as Python and C/C++, and with the help of hardware libraries/overlays, using a hardware–software codesign approach. The hardware libraries or overlays are written in conventional languages, such as Verilog, and can be used directly to program the FPGA on board the PYNQ-Z1 board. Here, we choose the PYNQ-Z1 board as it has very low power consumption, which makes it suitable for IoT and industrial applications. To monitor the power consumption of the PYNQ-Z1 board in real time, we used a USB power tester. A real-time image of the setup is given in Fig. 7. The PYNQ-Z1 board has a 650-MHz dual core Cortex-A9 processor and PL equivalent to Artix-FPGA. It has 13 300 logic slices, each with four six-input LUTs and eight flip flops. Details of its PL resources are given in Table III.

In addition to this, we also perform worst-case power estimation using the Xilinx Power Estimator (XPE) tool in the following section.

A. Power Estimation Using XPE

It is important to determine the power and cooling specifications of system on chip and FPGA designs early. Accurate worst-case power analysis helps in avoiding major roadblocks

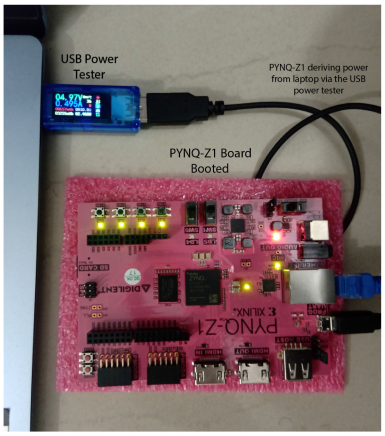
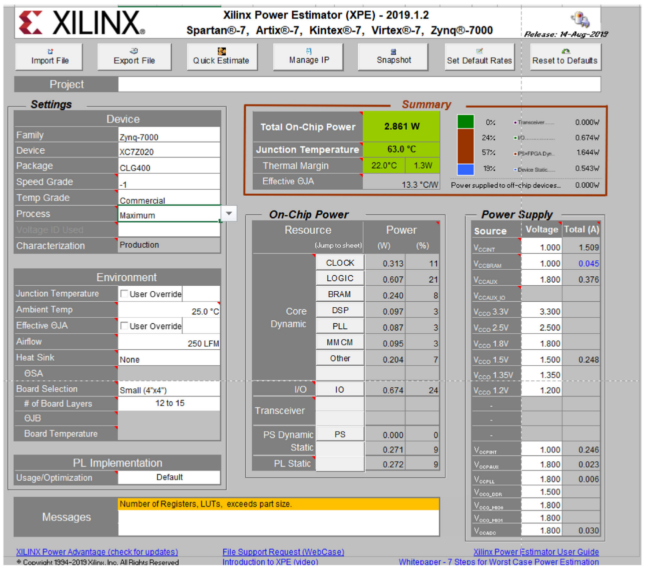


Fig. 7. PYNQ-Z1 board booted with the PYNQ image and drawing power from USB port via USB power tester.


XILINX
Xilinx Power Estimator (XPE) - 2019.1.2
Spartan®-7, Artix®-7, Kintex®-7, Virtex®-7, Zynq®-7000
Release: 14-Aug-2019
Import File
Export File
Quick Estimate
Manage IP
Snapshot
Set Default Rates
Reset to Defaults
Project
XILINX Power Advantage (check for updates)
File Support Request (WebCase)
Xilinx Power Estimator User Guide
Introduction to XPE (video)
Whitepaper - 7 Steps for Worst Case Power Estimation

Settings

Device	
Family	Zynq-7000
Device	XC7Z020
Package	CLG400
Speed Grade	-1
Temp Grade	Commercial
Process	Maximum
Voltage ID Used	
Characterization	Production

Environment	
Junction Temperature	☐ User Override
Ambient Temp	25.0 °C
Effective ΘJA	☐ User Override
Airflow	250 LFM
Heat Sink	None
ΘSA	
Board Selection	Small (4"x4")
# of Board Layers	12 to 15
ΘJB	
Board Temperature	

PL Implementation	
Usage/Optimization	Default

Summary

Total On-Chip Power	2.861 W	
Junction Temperature	63.0 °C	
Thermal Margin	22.0°C	1.3W
Effective ΘJA	13.3 °C/W	

🟩	0%	Transceiver......	0.000W
🟫	24%	I/O...............	0.674W
🟫	57%	PS+FPGA Dyn..	1.644W
🟦	19%	Device Static.....	0.543W
		Power supplied to off-chip devices...	0.000W

On-Chip Power

Resource	(Jump to sheet)	Power (W)	(%)
Core Dynamic	CLOCK	0.313	11
	LOGIC	0.607	21
	BRAM	0.240	8
	DSP	0.097	3
	PLL	0.087	3
	MMCM	0.095	3
	Other	0.204	7
I/O	IO	0.674	24
Transceiver			
PS Dynamic	PS	0.000	0
Static		0.271	9
PL Static		0.272	9

Power Supply

Source	Voltage	Total (A)
V_{CCINT}	1.000	1.509
V_{CCBRAM}	1.000	0.045
V_{CCAUX}	1.800	0.376
V_{CCAUX_IO}		
V_{CCO} 3.3V	3.300	
V_{CCO} 2.5V	2.500	
V_{CCO} 1.8V	1.800	
V_{CCO} 1.5V	1.500	0.248
V_{CCO} 1.35V	1.350	
V_{CCO} 1.2V	1.200	
-		
-		
-		
V_{CCPINT}	1.000	0.246
V_{CCPAUX}	1.800	0.023
V_{CCPLL}	1.800	0.006
V_{CCO_DDR}	1.500	
V_{CCO_MIO0}	1.800	
V_{CCO_MIO1}	1.800	
V_{CCADC}	1.800	0.030

Messages: Number of Registers, LUTs, exceeds part size.

® Copyright 1994-2019 Xilinx, Inc. All Rights Reserved

Fig. 8. Screenshot of the XPE Window for estimating the worst-case power consumption on the PYNO-Z1 board.

in the product development life cycle. The Xilinx SPE is a spreadsheet tool designed for this purpose. We performed the power estimation using the XPE for Zynq 7000 family (PYNQ Z1 board belongs to ZYNQ XC7Z020-1CLG400 C) [60], [61]. On choosing the suitable device settings and 100% resource utilization in the PL, the total on-chip power was estimated to be 2.861 W, as shown in Fig. 8. This proves the lower power consumption of the board and makes it feasible for checking designs meant for IoT deployment. This is also very close to the results obtained with the USB power tester, as discussed in Tables IV and V. Also, we plot the power consumption estimate using the XPE for different degrees of PL utilization from 20% to 100%, as shown in Fig. 9.

B. Adopted Methodology for FPGA and MCU Implementation

Next, we followed the following step-by-step methodology for the FPGA and MCU implementation of the various ML

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
COMPARISON OF RESULTS FOR EXPERIMENTS ON THE XILINX PYNQ-Z1 BOARD(PS) AND THE INTEL CORE CPU FOR TEMPERATURE DRIFT COMPENSATION

Model	Metrics for Xilinx PYNQ-Z1 Board(PS)					Metrics for Intel(R) Core(TM) i5-8250U CPU @ 1.6 GHz - Quad core				
	MAE	MSE	R^2	Time (s)	Power (W)	MAE	MSE	R^2	Time (s)	Power (W)
LR	0.234	1.237e-01	0.99121	**0.004**	**2.4552**	0.234	1.237e-01	0.99121	**∼0**	**6.0**
PR	0.071	1.118e-02	0.99916	0.013	2.7181	0.071	1.118e-02	0.99916	0.0009	9.7
DT	0.058	6.713e-03	0.99952	0.023	2.7181	0.058	6.713e-03	0.99952	0.0029	9.6
RF	**0.007**	**8.947e-05**	**0.99999**	0.272	2.7280	**0.007**	**8.627e-05**	**0.99999**	0.0568	10.8
MLP	0.025	9.809e-04	0.99993	0.035	2.52464	0.025	9.809e-04	0.99993	0.0366	9.6
SVR	0.074	1.060e-02	0.99925	73.878	2.66352	0.074	1.059e-02	0.99925	1.1847	21.2

TABLE V
COMPARISON OF RESULTS FOR EXPERIMENTS ON THE XILINX PYNQ-Z1 BOARD(PS) AND THE INTEL CORE CPU FOR TEMPERATURE-TEMPORAL DRIFT COMPENSATION

Model	Metrics for Xilinx PYNQ-Z1 Board(PS)					Metrics for Intel(R) Core(TM) i5-8250U CPU @ 1.6 GHz - Quad core				
	MAE	MSE	R^2	Time (s)	Power (W)	MAE	MSE	R^2	Time (s)	Power (W)
LR	1.4217	3.5449	0.642554	**0.0051**	**2.485**	1.4217	3.5449	0.642554	**0.0009**	9.6
PR	0.0577	0.0220	0.997777	0.0751	2.743	0.0581	0.0220	0.997776	0.0069	11.6
DT	0.0057	0.0005	0.999943	0.0229	2.847	0.0057	0.0005	0.999950	0.0019	**5.5**
RF	**0.0017**	**0.0003**	**0.999965**	0.4405	2.768	**0.0018**	**0.0003**	**0.999966**	0.0419	6.5
MLP	0.0552	0.0151	0.998472	0.0994	2.599	0.0552	0.0151	0.998473	0.0049	12.3
SVR	0.0704	0.8175	0.991757	60.636	2.644	0.0705	0.0818	0.991753	0.9520	13.4

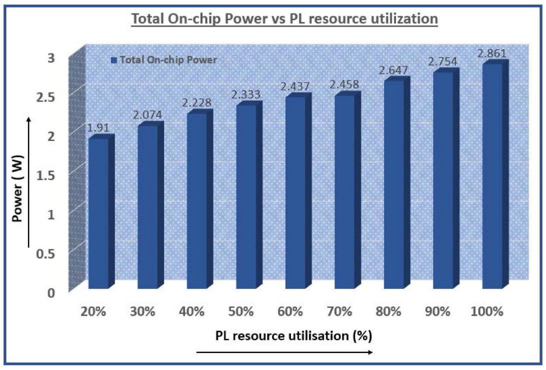


Fig. 9. Total on chip power consumption on the PYNQ Z1 board for different amount of resource utilization of the PL (estimated using XPE).

algorithms discussed earlier for temperature and temperature–temporal drift compensation in ISFET devices.

1) We installed the SciKit Learn (SK Learn) library on the PYNQ Z1 board using Andrei [62].
2) Next, we trained the ML models using the dataset generated from SPICE.
3) Next, we dump the trained model on the PS of PYNQ-Z1 and run the prediction. While the prediction is running, we measure the peak power consumption of the board by visually observing the USB power tester display.
4) We implement the ML algorithms for temperature and temporal drift data on the PL of the PYNQ-Z1 board using the Sklearn hardware library for PYNQ in [63].
5) We calculate the time taken for prediction using the time.time() function in the Jupyter terminal of the PYNQ Z1 board.
6) The performance metrics, such as MAE, MSE, RMS, and R^2, are calculated for the ML models using the Python code in the Jupyter terminal.
7) We note the obtained values and multiply the corresponding elapsed time with peak power consumption values to calculate the energy consumption per prediction.
8) Steps 2–6 are repeated for all the six ML algorithms for both temperature and temperature–temporal drift compensation.
9) As the best results were achieved by RF, we choose to implement it from scratch using Verilog and implemented it using Xilinx Vivado software. We selected the Xc7z020CLG400-1 FPGA platform (PL of PYNQ-Z1) for implementation. The Verilog code was extracted for the trained RF model from Snyder [64].
10) As RF consists of multiple DTs, we also implemented DT using the Xilinx Vivado Software on PL platform. Overlays given in [65] can also be used to run DT on the PYNQ-Z1 board directly.
11) Finally, all the results are compiled and presented in tabular format. The PL implementation of PYNQ Z1 is compared with the Intel Core CPU implementation in terms of accuracy, power consumption, and speed.
12) We calculate the new parameter proposed in Section V-E called IF score for the six ML algorithms. It indicates the best performing ML model for IoT deployment, giving equal importance to accuracy, power consumption, and time taken for prediction for both the PS and PL of the PYNQ-Z1 FPGA.

Recorded video and code snippets from Jupyter terminal of the PYNQ-Z1 board for the experiments conducted on the PS are given in the supplementary material [66].

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

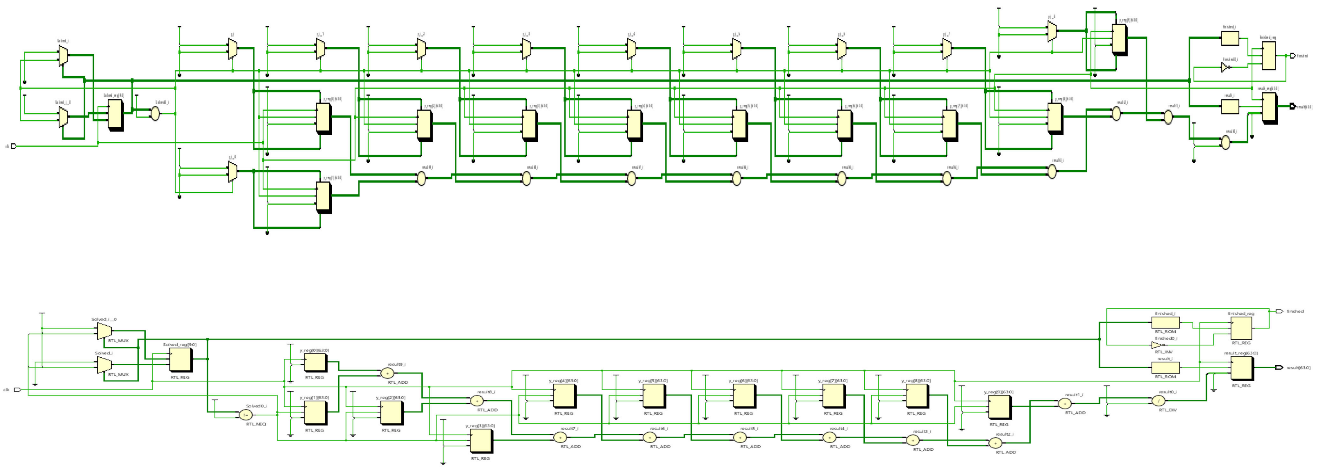


Fig. 10. Schematic of the trained RF algorithm implementation on the PL of PYNQ-Z1 board using generated using Xilinx Vivado software for only temperature (top) and temperature-temporal drift (bottom).

C. Issues With Fixed Point and Floating Point Representation

In FPGA implementation, it is important to decide the number representation to be used. There are two popular methods: 1) fixed point representation, and 2) floating point representation. In fixed point representations, it is easier to perform arithmetic operations but the range of the numbers and precision is limited, whereas in floating point representation (such as IEEE 754), it is difficult to perform arithmetic operations but the range of numbers, which can be represented, and the precision is very high. In our implementation of ML models on the PL, we used 32-b fixed point number representation with 20 b for fraction part. A better solution can be obtained by using standard floating point representation, such as IEEE 754 single precision, half precision, etc., which is part of our future work.

D. Details of FPGA Implementation

For LR and PR models, we used a common hardware library from the work in [63]. The library used 32-b fixed point number system with 20 b in fraction. For LR, the training:testing ratio in the dataset was 80:20 for temperature and temporal drift compensation. However, for PR, the same was 95:5 due to the limited buffer size available on the PYNQ-Z1 board for storing the test data. We also implemented the best performing RF model along with DT algorithms using the Xilins Vivado Software on the PYNQ-Z1 PL platform, i.e., XC7020CLG400-1. The Verilog code for the trained RF and DT models was extracted using Snyder [64]. It uses 64-b number representation with only 8 b for fractions. For both temperature and temperature–temporal drift compensation, the RF had 12 estimators with a maximum depth of 4. Fig. 10 shows the schematic of the trained RF models for temperature and temperature–temporal drift obtained using the Xilinx Vivado Software. For DT, in both temperature and temperature–temporal drift compensation, the depth was 4.

V. Results and Discussion

A. Temperature Drift Compensation

Table IV encompasses Si_3N_4-based ISFET ROIC results on various performance measures for achieving temperature drift compensation on both the Xilinx PYNQ-Z1 board and the Intel(R) Core(TM) i5-8250 CPU @ 1.6 GHz - Quad core. The developed ML models achieved very high R^2 score (correlation) on both the platforms. The strong correlation value between the real pH value and its expected value indicates that the temperature drift can be precisely accounted for by the ML models. The error rates achieved with various models are very small, with RF outperforming other models. We suspect that with a large number of data samples used for training, MLP might outperform the classical ML algorithms. In terms of the time taken for prediction and average power consumption during prediction, LR outperforms other models on both the platforms but gives poorer performance metrics as compared to other models. Thus, there is a tradeoff between better performance metrics with power and time consumed for prediction.

In addition, we examine the performance of the RF model with respect to a simple curve fitting model (we chose LR for this purpose). Illustration Fig. 11 shows the expected pH versus temperature curves for pH values 7 and 10. The obtained curves are almost same for both CPU and FPGA implementation. This is explained by results shown in Table IV, where it is seen that the performance metrics of FPGA implementation on PYNQ board and CPU implementation are similar for RF and LR models. The curves in blue correspond to an RF regressor fitted on all samples of training data, except for a number of pH values. For example, all samples corresponding to pH 0–5 and 9–14, i.e., except all of the samples corresponding to pH 6–8, will be the training data to predict pH 7. On similar lines, another model was trained to predict pH 10.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

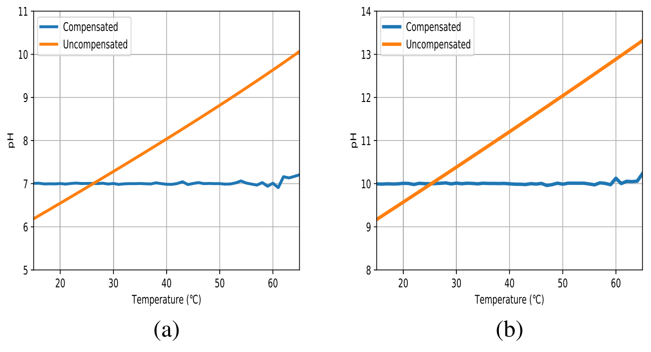


Fig. 11. Prediction curve with and without ROIC temperature drift compensation for pH 7 and pH 10 [1]. (a) pH-7. (b) pH-10.

B. Temperature–Temporal Drift Compensation

Table V shows the performance of ML models for compensating drift in V_{ref} due to both time and temperature using the Xilinx PYNQ-Z1 board and the Intel(R) Core(TM) i5-8250 CPU @ 1.6 GHz - Quad core. All ML models except LR perform decently well for the task at hand. This drop in performance of LR model can be explained due to the nonlinearity of the experiment. Even though it consumes the least time in both platforms and least power on FPGA board, its poor accuracy makes it unfit for implementation. Similar to experiment 1, a high R^2 score shows prominent correlation between the real V_{ref} and the predicted V_{ref} for particular temperature, pH, and time. RF achieves better results than any other model, showing its worth for compensating drift in voltage for Si_3N_4-based ISFET.

Moreover, here we examine the performance of the RF model for temperature–temporal drift compensation with respect to best model obtained from experiment 1 (only temperature compensation). Illustration Fig. 12 shows the expected pH versus time curves for pH values 7 and 10 at two temperatures (low and high). For training the RF regressor, for both temporal and temperature drift compensation, it was fitted on all samples of training data except for a number of temperatures. For example, if the model is used to compensate V_{ref} for pH 7 at 20°C, it was trained for all samples of pH 7 and all temperatures except 18,19,20,21,22°C. On similar lines, another model was trained to predict V_{ref} for pH 10. The output is mapped to corresponding pH value. Similar to the previous case, the results are nearly same for both the FPGA and CPU implementations due to the very close performance metrics, as shown in Table V.

C. Time Series Classification

We compare the performance on four different RNN-based models—1) LSTM, 2) biLSTM, 3) GRU, and 4) biGRU [67]. The number of RNN nodes in LSTM and GRU is 32, whereas in biLSTM and biGRU is 16. The feed-forward NN has 200 neurons, which is chosen based on grid search in the set 10,20,50,100,150,200,300,500,1000. We minimize the binary cross-entropy loss with a batch size of 64 and run for 10 epochs. Table VI shows biLSTM performs best amongst all the RNN variants with minimum entropic loss and highest accuracy of

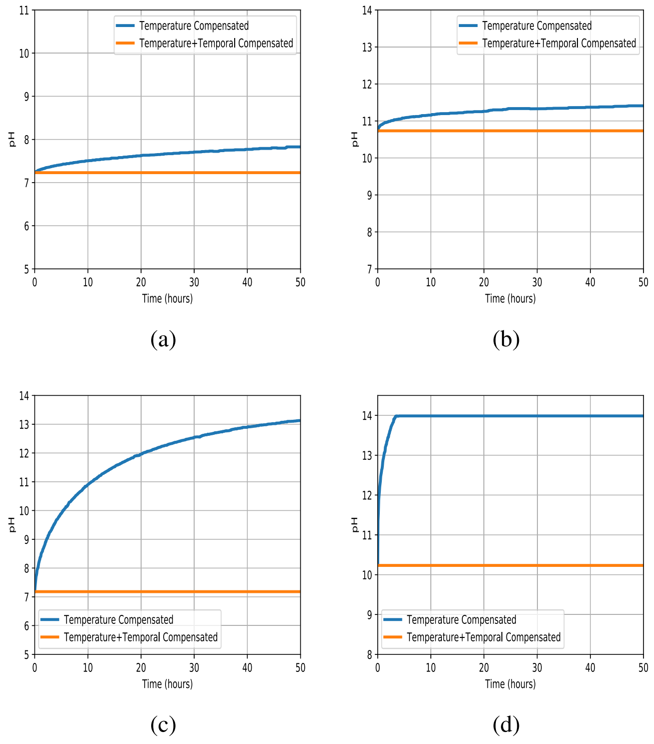


Fig. 12. Prediction curve with both temporal and temperature drift compensation and only temperature compensation. (a) pH-time curve for pH 7, T = 20 °C. (b) pH-time curve for pH 10, T = 20 °C. (c) pH-time curve for pH 7, T = 35 °C. (d) pH-time curve for pH 10, T = 35 °C [1].

TABLE VI
COMPARISON OF RESULTS OBTAINED FOR DIFFERENT RNN ON CPU FOR TEMPERATURE DRIFT COMPENSATION

RNN-model	Loss	Accuracy
LSTM	0.4135	79.20%
GRU	0.4061	79.00%
BiLSTM	0.4063	79.80%
BiGRU	0.486	79.20%

79.8%. As the obtained accuracy of RNN is not very high in CPU, future work is required to optimize it further before going for FPGA implementation. As a future work, when the data scarcity is not a limitation, the classification task can be effortlessly adapted to a regression task, where a sequence of voltages can predict precise pH of the solution over a wide range.

D. Comparison of FPGA, MCU, and CPU Implementation

In this section, we compare the FPGA, MCU, and CPU implementation of the various ML algorithms in terms of three parameters, i.e., accuracy, speed, and power consumption. Here, FPGA implementation is on the PS+PL of the PYNQ-Z1 board, MCU implementation is only on the PYNQ-Z1 board, and CPU implementation is on the Intel Quad core processor.

1) Accuracy: Accuracy is measured in terms of performance metrics, i.e., MAE, MSE, and correlation (R^2). For an accurate model, MAE and MSE should be very low and R^2 should be very

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
PERFORMANCE METRICS FOR ML ALGORITHMS PRACTICALLY IMPLEMENTED USING THE PYNQ-Z1 BOARD PL (XC7Z020CLG400-1) AND POWER MEASURED USING USB POWER TESTER

Drift Compensation	ML Algorithm	Accuracy Metrics			Time taken (per prediction)	Energy consumed (per prediction) (J)	IF Score
		MAE	MSE	R2			
Only Temperature	LR	0.244	0.125	0.991	3.80E-04	6.83E-04	1351.861
	PR	0.015	0.003	0.999	4.24E-04	7.70E-04	1248.598
Temperature + Temporal	LR	1.505	3.963	0.644	3.68E-04	6.24E-04	1425.462
	PR	0.695	1.001	0.907	3.67E-04	6.50E-04	1407.147

TABLE VIII
PERFORMANCE METRICS FOR TRAINED RF MODEL IMPLEMENTATIONS ON VIVADO ON XC7Z020CLG400-1 FPGA BOARD (PL OF PYNQ-Z1)

Drift Comp.	ML Algorithm	Accuracy Metrics			Total on chip power (W)
		MAE	MSE	R2	
Only Temperature	RF	0.598	0.546	0.961	1.13
	DT	0.692	0.7248	0.948	1.13
Temperature + Temporal	RF	0.297	0.335	0.965	0.926
	DT	0.316	0.397	0.958	1.13

TABLE IX
OVERALL COMPARISON OF FPGA AND CPU IMPLEMENTATION OF ML MODELS FOR ISFET IOT AND INDUSTRIAL IMPLEMENTATION

Parameter	Parameter requirement for ISFET IoT and Industrial applications	FPGA implementation (PL of PYNQ-Z1 Board)	MCU implementation (PS of PYNQ-Z1 Board)	CPU implementation
Accuracy	Essential	✓	=	=
Speed	Non-Essential but helpful	✓	✗	✓
Power	Essential	✓	✓	✗

high. We see in Tables IV, V, VII, and VIII that the accuracy of all the three, the CPU, MCU, and FPGA, implementation is nearly the same. FPGA has slightly lower accuracy but that can be easily overcome by using more number of bits and floating point number representation.

2) Speed: Speed refers to the time duration ML model takes to predict the pH value. The lesser is the time taken, faster is the speed. From Tables IV, V, and VII, we observe that the speed is much better for the CPU implementation in both temperature and temperature–temporal compensation tasks as compared to MCU implementation. It should be noted that the FPGA implementation is faster than both in most cases. Hence, FPGA helps in achieving hardware acceleration.

3) Power Consumption: Power is measured while the ML models are running on the FPGA, MCU, and the CPU. The lesser is the power consumption, better is the model. Here, for both the PS and PL, we measure the entire power consumption by the PYNQ-Z1 board with the help of a USB power tester, as shown in Fig. 7, whereas for the CPU, we measure the total power consumed by the CPU package, CPU cores, CPU graphics, and CPU DRAM with the help of the Open Hardware Monitor application. It is important to note that the PYNQ-Z1 board has many on-board peripherals, such as audio/video, USB, Ethernet, UART, CAN controllers, etc., which will not be required for final FPGA deployment in an IoT or industrial scenario for ISFETs, hence the power consumed by the ML models solely on the FPGA will be much lesser than what is given, i.e., the given power consumption can be treated as worst case. However, for the CPU, we are measuring the power consumed by the CPU itself and not other peripherals of the computer, hence it is the best case. It can be seen from Tables IV and V that the worst-case power consumed by FPGA implementation is much lesser than the best-case power consumption by CPU. Similarly, from Table VII, it is seen that the FPGA implementation provides a lot of hardware acceleration as compared to MCU or software implementation. Additionally, power consumption can also be measured using XPE or Xilinx Vivado Design Suite.

IoT and industrial scenarios, such as biomedical applications, environmental monitoring, agriculture, food industry, etc., where ISFETs are used may not require high speed but require low power consumption and high accuracy. Hence, FPGA implementation is the best approach for such applications. In this study, we used the PYNQ framework to implement the algorithms on FPGA but designs implemented using these are not optimized for performance parameters, such as speed, power consumption, area, etc. Hence, customized RTL and/or ASIC implementations would consume much less power and would be preferred for eventual large-scale fabrication. It will give better results than the results obtained in this study. Table IX summarizes the pros and cons of both the implementations. After concluding that the FPGA implementation is better than CPU or MCU for deployment of ML model in IoT and industrial application of ISFETs, the next step is to evaluate the best ML model to be deployed on FPGA. This is discussed in the next section.

E. Evaluation of ML Models for FPGA and MCU Hardware for IoT Deployment

As discussed previously, certain ML models, such as RF, give very high accuracy, whereas some ML models, such as LR, consume very less power and time but give relatively poorer accuracy and some models, such as DT, give very moderate accuracy and consume moderate power and time. Hence, choosing the optimum ML model for implementation on FPGA/MCU needs further analysis. In order to solve this problem, we define a new parameter called IF, which gives weights to accuracy, speed, or time taken per-prediction and energy consumed per-prediction to calculate a final score, which allows the ML models to be compared. The weights are given according to the following scheme.

1) Accuracy: 33% (11%—MAE, 11%—MSE, 11%—R^2).
2) Speed or time taken per prediction: 33%.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

TABLE X
RF- Final Comparison of Various ML Algorithms Based on the Accuracy of Prediction, Time Taken for Prediction, and Energy Consumed Per Prediction

Model	Temperature Drift Compensation						Temperature-Temporal Drift Compensation					
	Accuracy			Time (s)	Energy per prediction (mJ)	IF Score	Accuracy			Time (s)	Energy per prediction (mJ)	IF Score
	MAE	MSE	R^2				MAE	MSE	R^2			
LR	0.234	1.237e-01	0.99121	**0.004**	**9.921**	117.22	1.4217	3.5449	0.642554	**0.0051**	**12.817**	89.91
PR	0.071	1.118e-02	0.99916	0.013	36.654	44.39	0.0577	0.0220	0.997777	0.0751	205.906	13.01
DT	0.058	6.713e-03	0.99952	0.023	63.698	37.66	0.0057	0.0005	0.999943	0.0229	**65.349**	235.33
RF**	**0.007**	**8.947e-05**	**0.99999**	0.272	743.599	**1246.07**	**0.0017**	**0.0003**	**0.999965**	0.4405	1219.228	**378.85**
MLP	0.025	9.809e-04	0.99993	0.035	88.908	129.74	0.0552	0.0151	0.998472	0.0994	258.366	13.96
SVR	0.074	1.060e-02	0.99925	73.878	196777.458	11.96	0.0704	0.8175	0.991757	60.636	160303.618	1.81

Note: The IF score gives the overall performance score on the basis of which RF is best for implementation on FPGA platform for both temperature and temperature-temporal compensation.

TABLE XI
Comparison of Drift Compensation by ML Implemented on Platforms, Such as FPGA/MCUs Versus Other Drift Compensation Techniques

Comparison Criteria	Compensation by sensor design	Compensation using simple modelling	Compensation using post data processing techniques like ML on platforms like FPGA
Hardware implementation	Very complex as precise operations need to be performed for adjusting appropriate doping to compensate the temporal drift at fabrication level by ion-implantation. Also, it is highly vulnerable to process variation which may adversely impact the compensation [10].	Difficult to implement the modeling based compensation in hardware as modeling software like SPICE and MATLAB use complex mathematical tools and equations which are difficult to be implemented in IoT Embedded systems [68, 69].	ML models can be easily implemented in external hardware processing chips like FPGAs and trained to learn the drift patterns using the well-established ML models.
Size	No external chip based compensation required, hence overall system is small in size.	System size increases due to involvement of large and bulky computing systems.	System size slightly increases due to an additional chip.
Cost	Very costly as there has to be customization at the fabrication level.	Less costly	Less costly
Time	Fast, as there is compensation at the point of sensing.	Slow, as the data will have to be transferred to computing systems and calculated by complex softwares.	Fast, as it can have high level of parallelism.
Power	Not much power consumption.	High power consumption by computing systems.	Low power consumption.
Overall IoT deployment scenario	Feasible for Real-Time IoT Applications but will be very costly and cannot be reconfigured.	Implementation is challenging for real-time IoT applications and it may not be reconfigurable.	Feasible for real-time IoT applications and it will be cheap and reconfigurable

3) Energy consumed per prediction: 33%.

The IF score for each ML model can be calculated using the following formula:

$$\mathrm{IF} = \frac{0.11}{\mathrm{MAE}} + \frac{0.11}{\mathrm{MSE}} + 0.11 \times R^2 + \frac{0.33}{T} + \frac{0.33}{E} \quad (8)$$

where MAE, MSE, and R^2 are the performance metrics for accuracy of the ML model, T is the time taken by the ML model per prediction in SI unit, i.e., seconds, and E is the energy consumed by the ML model per prediction in SI unit, i.e., joules. The higher is the IF score, better is the ML model for implementation on FPGA/MCU, giving equal weights to all the three parameters, i.e., accuracy, speed, and energy consumption. To the best of our knowledge, there is no parameter defined in the literature till now which considers all the three factors: energy consumed, speed, and accuracy to give a single score to evaluate the best ML algorithm for FPGA/MCU implementation, which necessitates the creation of new IF parameter. Existing performance metrics for ML algorithms consider only one of these metrics, for example, F score and F1 score only gives information about the accuracy. Hence, there is a need to define a score which accounts collectively for all three factors. Moreover, the weights assigned to the factors may change according to type of application. For example, in bio-medical and research applications, accuracy will have more weight, whereas for long-term IoT deployment, such as environmental monitoring, power consumption will have more weight. In this study, as a general case, we have given equal weights to all the three factors. On the basis of IF score calculation, we see from Table X that RF has the highest IF for both temperature as well as temperature–temporal drift compensation. This is also graphically shown in Fig. 13.

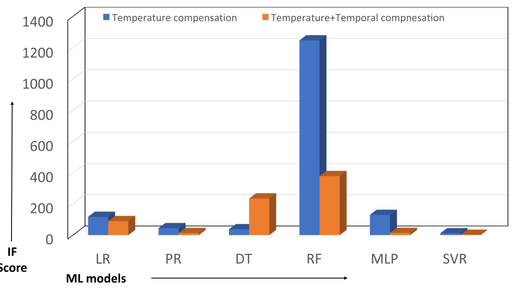


Fig. 13. IF score comparison of the various ML models for both temperature and temperature–temporal compensation.

F. Advantage of Drift Compensation Using ML Implemented on FPGA/MCU Platforms

As compared to conventional drift compensation methods in ISFETs, implementing ML on FPGA/MCUs has several advantages. Drift compensation in ISFETs may be done by other compensation techniques, such as ISFET sensor design or simple modeling. However, the temporal drift compensation is a complex problem, where the sensor needs to compensate for the drift temporally for various pH values. In order to learn the sensor behavior, ML models are suitable to capture the drift patterns in the acidic and basic pH solutions, and predict more accurate measurements in comparison to simple modeling techniques. Moreover, the ML models developed in this work are designed to obtain early estimation of pH even before the V_{Ref} stabilizes. Thus, its prediction can be much faster. We have discussed the advantages and limitations of using ML models implemented on FPGA platforms for sensor drift compensation in comparison with drift compensation by sensor design or using simple modeling, as listed in Table XI. Drift compensation using

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

ML implemented on FPGA platforms is suitable for industrial IoT applications, as it is reconfigurable and cheaper compared to other techniques.

VI. Conclusion

In this work, we present the modeling of nonidealities in Si_3N_4-gate ISFET due to temperature and temporal variations using SPICE and the drift compensation using state-of-the-art ML models, which have been implemented on FPGA. The SPICE macromodel is used as a subcircuit block in a CVCC readout topology to generate temperature and temporal drift data for a wide temperature (15–65°C) and pH (0–14) range. The data are used to train state-of-the-art ML models to compensate the temperature and temporal drift and it was found that RF technique outperforms other ML models. After observing promising results on CPU, the ML models were implemented in PYNQ Z1 board that houses both an ARM processor and an FPGA. It is a power and memory constrained environment suitable for IoT applications. We used hardware–software codesign based tool, PYNQ framework, to implement the design but customized RTL and/or ASIC implementations would consume much less power and would be preferred for eventual large-scale fabrication. We also propose a novel IF, which gives equal weights to accuracy, speed, and energy consumption while running on FPGA and assigns the ML models an overall score. A higher IF score indicates that the concerned ML model is better suited for FPGA implementation. Based on the evaluation, we find that RF has a higher IF score for both temperature and temperature–temporal drift compensation, in both the PS and PL implementations on the PYNQ-Z1 board. This article presents a promising application of ML in drift compensation of ISFET-based pH sensor, which has a huge scope for further research to improve the robustness of smart chemical sensors, which can be deployed for various industrial IoT applications, such as biomedical diagnosis, environmental monitoring, agriculture, wearable sensing, food industry, and horticulture.

Acknowledgment

The authors would like to thank the Director, CSIR-CEERI, Pilani, for the generous support. They would also like to thank all the scientific and technical staff of CSIR-CEERI, Pilani, for the discussions and technical support.

References

[1] A. Mehta, H. Ahuja, N. Sahu, R. Bhardwaj, S. Srivastava, and S. Sinha, “Machine learning techniques for performance enhancement of Si_3N_4-gate ISFET pH sensor,” in *Proc. IEEE 17th India Council Int. Conf.*, 2020, pp. 1–7.
[2] M. J. Madou and S. R. Morrison, *Chemical Sensing With Solid State Devices*. Amsterdam, The Netherlands: Elsevier, 2012.
[3] B. R. Eggins, *Chemical Sensors and Biosensors*, vol. 28. Hoboken, NJ, USA: Wiley, 2008.
[4] L. Manjakkal, D. Szwagierczak, and R. Dahiya, “Metal oxides based electrochemical pH sensors: Current progress and future perspectives,” *Prog. Mater. Sci.*, vol. 109, Apr. 2019, Art. no. 100635.
[5] M. Kaisti, “Detection principles of biological and chemical FET sensors,” *Biosensors Bioelectron.*, vol. 98, pp. 437–448, 2017.
[6] N. Moser, T. S. Lande, C. Toumazou, and P. Georgiou, “ISFETs in CMOS and emergent trends in instrumentation: A review,” *IEEE Sensors J.*, vol. 16, no. 17, pp. 6496–6514, Sep. 2016.
[7] N. Sahu, R. Bhardwaj, H. Shah, R. Mukhiya, R. Sharma, and S. Sinha, “Towards development of an ISFET-based smart pH sensor: Enabling machine learning for drift compensation in IoT applications,” *IEEE Sensors J.*, vol. 21, no. 17, pp. 19013–19024, Sep. 2021.
[8] S. Sinha *et al.*, “Modeling and simulation of temporal and temperature drift for the development of an accurate ISFET SPICE macromodel,” *J. Comput. Electron.*, vol. 19, no. 1, pp. 367–386, 2020.
[9] P. Barabash, “Analysis of the threshold voltage and its temperature dependence in electrolyte-insulator-semiconductor field-effect transistors (EISFET’s),” *IEEE Trans. Electron Devices*, vol. ED-34, no. 6, pp. 1271–1282, Jun. 1987.
[10] A. Elyasi, M. Fouladian, and S. Jamasb, “Counteracting threshold-voltage drift in ion-selective field effect transistors (ISFETs) using threshold-setting ion implantation,” *IEEE J. Electron. Devices Soc.*, vol. 6, no. 1, pp. 747–754, Jun. 2018.
[11] V. Pachauri and S. Ingebrandt, “Biologically sensitive field-effect transistors: From ISFETs to NanoFETs,” *Essays Biochem.*, vol. 60, no. 1, pp. 81–90, 2016.
[12] N. Zimmerman *et al.*, “A machine learning calibration model using random forests to improve sensor performance for lower-cost air quality monitoring,” *Atmos. Meas. Techn.*, vol. 11, no. 1, pp. 291–313, 2018.
[13] A. Moraru, M. Pesko, M. Porcius, C. Fortuna, and D. Mladenic, “Using machine learning on sensor data,” *J. Comput. Inf. Technol.*, vol. 18, no. 4, pp. 341–347, 2010.
[14] S. Sinha, R. Bhardwaj, N. Sahu, H. Ahuja, R. Sharma, and R. Mukhiya, “Temperature and temporal drift compensation for Al_2O_3-gate ISFET-based pH sensor using machine learning techniques,” *Microelectron. J.*, vol. 97, 2020, Art. no. 104710.
[15] G. Seo *et al.*, “Rapid detection of COVID-19 causative virus (SARS-COV-2) in human nasopharyngeal swab specimens using field-effect transistor-based biosensor,” *ACS Nano*, vol. 14, no. 4, pp. 5135–5142, 2020.
[16] J. Zimmer, “Lab-on-a-chip (LoC) COVID-19 test advances to clinical trials,” 2020. Accessed: Nov. 17, 2020. [Online]. Available: https://new.engineering.com/story/lab on a-chip-loc-covid-19-test-advances-to-clinical-trials
[17] D. Sadighbayan and E. Ghafar-Zadeh, “Portable sensing devices for detection of COVID-19: A review,” *IEEE Sensors J.*, vol. 21, no. 9, pp. 10219–10230, May 2021.
[18] T. Shaffaf and E. Ghafar-Zadeh, “COVID-19 diagnostic strategies. Part I: Nucleic acid-based technologies,” *Bioengineering*, vol. 8, no. 4, pp. 1–29, 2021.
[19] X. Huang, H. Yu, X. Liu, Y. Jiang, M. Yan, and D. Wu, “A dual-mode large-arrayed CMOS ISFET sensor for accurate and high-throughput pH sensing in biomedical diagnosis,” *IEEE Trans. Biomed. Eng.*, vol. 62, no. 9, pp. 2224–2233, Sep. 2015.
[20] A. M. Dinar, A. M. Zain, and F. Salehuddin, “CMOS ISFET device for DNA sequencing: Device compensation, application requirements and recommendations,” *Int. J. Appl. Eng. Res.*, vol. 12, no. 21, pp. 11015–11028, 2017.
[21] P. Zarkesh-Ha, J. Edwards, and P. Szauter, “Avalanche ISFET: A highly sensitive pH sensor for genome sequencing,” in *Proc. IEEE Biomed. Circuits Syst. Conf.*, 2015, pp. 1–4.
[22] N. Miscourides and P. Georgiou, “Impact of technology scaling on ISFET performance for genetic sequencing,” *IEEE Sensors J.*, vol. 15, no. 4, pp. 2219–2226, Apr. 2015.
[23] A. Ganguli, Y. Watanabe, M. T. Hwang, J.-C. Huang, and R. Bashir, “Robust label-free microRNA detection using one million ISFET array,” *Biomed. Microdevices*, vol. 20, no. 2, pp. 1–10, 2018.
[24] G. Alexandrou *et al.*, “Detection of breast cancer ESR1 p.E380Q mutation on an ISFET lab-on-chip platform,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2020, pp. 1–5.
[25] M. Lehmann *et al.*, “Simultaneous measurement of cellular respiration and acidification with a single CMOS ISFET,” *Biosensors Bioelectron.*, vol. 16, no. 3, pp. 195–203, 2001.
[26] J.-H. Jeon and W.-J. Cho, “Ultrasensitive coplanar dual-gate ISFETs for point-of-care biomedical applications,” *ACS Omega*, vol. 5, no. 22, pp. 12809–12815, 2020.
[27] M. Kalofonou and C. Toumazou, “Semiconductor technology for early detection of DNA methylation for cancer: From concept to practice,” *Sensors Actuators B: Chem.*, vol. 178, pp. 572–580, 2013.
[28] J. Malinowski and E. J. Geiger, “Development of a wireless sensor network for algae cultivation using ISFET pH probes,” *Algal Res.*, vol. 4, pp. 19–22, 2014.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.

[29] G. A. Taylor, C. Parra, H. Carrillo, and A. Mouazen, "A decision framework reference for ISFET sensor-based electronic systems design for agriculture industry applications," in *Proc. IEEE 17th India Council Int. Conf.*, 2020, pp. 1–6.

[30] D. Walvoort, J. Bouma, P. Peters, and J. de Gruijter, "Using ISFETs for proximal sensing in precision agriculture," in *Proc. 5th Int. Conf. Precis. Agriculture*, 2000, pp. 1–11.

[31] G. A. Taylor *et al.*, "pH measurement IoT system for precision agriculture applications," *IEEE Latin Amer. Trans.*, vol. 17, no. 5, pp. 823–832, May 2019.

[32] C. Jimenez-Jorquera, J. Orozco, and A. Baldi, "ISFET based microsensors for environmental monitoring," *Sensors*, vol. 10, no. 1, pp. 61–83, 2010.

[33] M. Joly, L. Mazenq, M. Marlet, P. Temple-Boyer, C. Durieu, and J. Launay, "Multimodal probe based on ISFET electrochemical microsensors for in-situ monitoring of soil nutrients in agriculture," in *Proc. Multidisciplinary Digit. Publishing Inst.*, vol. 1, no. 4, pp. 1–4, 2017.

[34] S. J. Birrell and J. W. Hummel, "Real-time multi ISFET/FIA soil analysis system with automatic sample extraction," *Comput. Electron. Agriculture*, vol. 32, no. 1, pp. 45–67, 2001.

[35] S. J. Birrell and J. W. Hummel, "Membrane selection and ISFET configuration evaluation for soil nitrate sensing," *Trans. Amer. Soc. Agricultural Eng.*, vol. 43, no. 2, pp. 197–206, 2000.

[36] M. Joly, L. Mazenq, M. Marlet, P. Temple-Boyer, C. Durieu, and J. Launay, "All-solid-state multimodal probe based on ISFET electrochemical microsensors for in-situ soil nutrients monitoring in agriculture," in *Proc. 19th Int. Conf. Solid-State Sensors, Actuators, Microsyst.*, 2017, pp. 222–225.

[37] V. I. Adamchuk, J. W. Hummel, M. Morgan, and S. Upadhyaya, "On-the-go soil sensors for precision agriculture," *Comput. Electron. Agriculture*, vol. 44, no. 1, pp. 71–91, 2004.

[38] N. F. Starodub, "Efficiencies of biosensors in environmental monitoring," in *Portable Biosensing of Food Toxicants and Environmental Pollutants, Series in Sensors*, D. P. Nikolelis *et al.*, Eds. New York, NY, USA:Taylor & Francis, 2013, pp. 487–514.

[39] K. McLaughlin *et al.*, "An evaluation of ISFET sensors for coastal pH monitoring applications," *Regional Stud. Mar. Sci.*, vol. 12, pp. 11–18, 2017.

[40] N. Jaffrezic-Renault, A. Senillou, C. Martelet, K. Wan, and J. Chovelon, "ISFET microsensors for the detection of pollutants in liquid media," *Sensors Actuators B: Chem.*, vol. 59, no. 2/3, pp. 154–164, 1999.

[41] E. M. Briggs, S. Sandoval, A. Erten, Y. Takeshita, A. C. Kummel, and T. R. Martz, "Solid state sensor for simultaneous measurement of total alkalinity and pH of seawater," *ACS Sensors*, vol. 2, no. 9, pp. 1302–1309, 2017.

[42] E. Voulgari, F. Krummenacher, and M. Kayal, "ANTIGONE: A programmable energy-efficient current digitizer for an ISFET wearable sweat sensing system," *Sensors*, vol. 21, no. 6, pp. 1–16, 2021.

[43] S. Nakata, T. Arie, S. Akita, and K. Takei, "Wearable, flexible, and multifunctional healthcare device with an ISFET chemical sensor for simultaneous sweat pH and skin temperature monitoring," *ACS Sensors*, vol. 2, no. 3, pp. 443–448, 2017.

[44] M. Douthwaite, E. Koutsos, D. C. Yates, P. D. Mitcheson, and P. Georgiou, "A thermally powered ISFET array for on-body pH measurement," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 6, pp. 1324–1334, Dec. 2017.

[45] F. Bellando, L. J. Mele, P. Palestri, J. Zhang, A. M. Ionescu, and L. Selmi, "Sensitivity, noise and resolution in a BEOL-modified foundry-made ISFET with miniaturized reference electrode for wearable point-of-care applications," *Sensors*, vol. 21, no. 5, pp. 1–19, 2021.

[46] A. Vilouras and R. Dahiya, "Ultra-thin chips with current-mode ISFET array for continuous monitoring of body fluids pH," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2021, pp. 1–5.

[47] N. F. Starodub, W. Torbicz, V. M. Starodub, M. I. Kanjuk, and K. S. Ternovoj, "Enzymatic sensors based on the ISFETs for determination of heavy metal ions in some vegetables," in *Proc. Int. Solid State Sensors Actuators Conf.*, 1997, vol. 2, pp. 1383–1384.

[48] V. Volotovsky and N. Kim, "Determination of glucose, ascorbic and citric acids by two-ISFET multienzyme sensor," *Sensors Actuators B: Chem.*, vol. 49, no. 3, pp. 253–257, 1998.

[49] N. Starodub, Y. M. Shirshov, W. Torbicz, N. Kanjuk, V. Starodub, and A. Kukla, "Biosensors for in field measurements: Optimisation of parameters to control phosphororganic pesticides in water and vegetables," in *Biosensors for Direct Monitoring of Environmental Pollutants in Field*. Berlin, Germany: Springer, 1998, pp. 209–219.

[50] H. Ito *et al.*, "Nitrate ion determination of vegetables using a portable ISFET-nitrate ion sensor," *Bull. Nat. Inst. Vegetable Tea Sci.*, vol. 15, no. 15, pp. 11–17, 2016.

[51] H. Van den Vlekkert, J. P. M. Kouwenhoven, and A. Van Wingerden, "Application of ISFETs in closed-loop systems for horticulture," in *Proc. Int. Workshop Sensors Horticulture*, 1991, vol. 304, pp. 309–320.

[52] H. Van den *et al.*, "Multi-ion sensing device for horticultural application based upon chemical modification and special packaging of ISFETs," *Sensors Actuators B: Chem.*, vol. 6, no. 1–3, pp. 34–37, 1992.

[53] J. Artigas *et al.*, "Application of ion sensitive field effect transistor based sensors to soil analysis," *Comput. Electron. Agriculture*, vol. 31, no. 3, pp. 281–293, 2001.

[54] I. Amemiya, H. Yagi, and T. Sakai, "An ISFET-based nutrient sensor for rockwool culture," *IFAC Proc. Vol.*, vol. 24, no. 11, pp. 361–366, 1991.

[55] T. H. Gieling and H. Van Den Vlekkert, "Application of ISFETs in closed-loop systems for greenhouses," *Adv. Space Res.*, vol. 18, no. 4/5, pp. 135–138, 1996.

[56] S. Naimi, B. Hajji, Y. Habbani, I. Humenyuk, J. Launay, and P. Temple-Boyer, "Modeling of the pH-ISFET thermal drift," in *Proc. Int. Conf. Microelectron.*, 2009, pp. 288–291.

[57] S. Jamasb, S. Collins, and R. L. Smith, "A physical model for drift in pH ISFETs," *Sensors Actuators B: Chem.*, vol. 49, no. 1/2, pp. 146–155, 1998.

[58] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[59] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2020.

[60] Xilinx, "PYNQ: Python productivity." 2021, Accessed: May 18, 2021. [Online]. Available: http://www.pynq.io/

[61] Digilent, "PYNQ-Z1 Reference Manual." 2021, Accessed: Jul. 25, 2021. [Online]. Available: https://reference.digilentinc.com/programmable-logic/pynq-z1/reference-manual

[62] I. Andrei, "PYNQ-Z1 error use pip and install Sklearn." 2021, Accessed: Jul. 26, 2021. [Online]. Available: https://discuss.pynq.io/t/pynq-z1-error-use-pip-and-install-sklearn/129

[63] sfox14, "GitHub: Sfox/PYNQ Sklearn." 2021, Accessed: Aug. 15, 2021. [Online]. Available: https://github.com/sfox14/pynq-sklearn

[64] B. Snyder, "GitHub: Johnbensnyder/fpgarandomforest." 2021, Accessed: Aug. 15, 2021. [Online]. Available: https://github.com/johnbensnyder/FPGA_random_forest

[65] Y. Qu, "GitHub: Yunqu/pynq-decisiontree." 2021, Accessed: Aug. 15, 2021. [Online]. Available: https://github.com/yunqu/PYNQ-DecisionTree

[66] S. Sinha, N. Sahu, R. Bhardwaj, A. Mehta, and H. Ahuja, "Machine learning on FPGA for robust Si_3N_4-gate ISFET pH sensor in industrial IoT applications," 2021. [Online]. Available: https://dx.doi.org/10.21227/pvt8-3n83

[67] W. Yu, I. Y. Kim, and C. Mechefske, "Analysis of different RNN autoencoder variants for time series classification and machine prognostics," *Mech. Syst. Signal Process.*, vol. 149, 2021, Art. no. 107322.

[68] P. K. Chan and D. Chen, "A CMOS ISFET interface circuit with dynamic current temperature compensation technique," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 54, no. 1, pp. 119–129, Jan. 2007.

[69] W.-Y. Chung, C.-H. Yang, M.-C. Wang, D. G. Pijanowska, and W. Torbicz, "Temperature compensation electronics for ISFET readout applications," in *Proc. IEEE Int. Workshop Biomed. Circuits Syst.*, 2004, pp. S1–S6.

Authorized licensed use limited to: BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE. Downloaded on July 09,2024 at 14:46:05 UTC from IEEE Xplore. Restrictions apply.