Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.
Digital Object Identifier XX.XXXX/YYY.20XX.ZZZZZZZ

Explainable Neuro-Memristive Systems

RAHUL KOTTAPPUZHACKAL[1], SRUTHI PALLATHUVALAPPIL[2], AND ALEX JAMES [3]
[1]School of Electronic Systems and Automation, Digital University Kerala, Trivandrum, India
Corresponding author: Alex James (e-mail: e-mail: apj@ieee.org

ABSTRACT
System level simulation of neuro-memristive circuits under variability are complex and follow a black-box neural network approach. In realistic hardware, they are often difficult to cross-check for accuracy and reproducible results. The accurate memristor model prediction becomes critical to decipher the overall the circuit function in wide range of non-ideal and practical conditions. In most neuro-memristive systems, crossbar configuration is essential for implementing multiply and accumulate calculations, that forms the primary unit for neural network implementations. Predicting the specific memristor model that best fits the crossbar simulations to make it explainable is an open challenge that is solved in this paper. As the size of the crossbar increases the cross-validation becomes an even more challenging. This paper proposes predicting the memristor device under test by automatically evaluating the I-V behavior using Random forest and Extreme Gradient Boosting algorithms. Starting with a single memristor model, the prediction approach is extended to memristor crossbar-based circuits explainable. The performance of both algorithms is analysed based on precision, recall, f1-score and support. The accuracy, macro average and weighted average of both algorithms at different operational frequencies are explored.

INDEX TERMS Memristor models, Memristor crossbar, Pinched hysteresis, Random forest predictor, Extreme Gradient Boost predictor

I. INTRODUCTION

MEMORY resistors are a class of devices abbreviated as memristors [1]. It is the fourth basic circuit element that functionally relates the charge and linkage flux. Their properties differ from the other three fundamental devices by their non-volatile memory effect, pinched hysteresis loop, scalability, programming capability, and compatibility with CMOS technology. It memorises the latest attained conductance value even if the power supply is off. Due to these features, the application of memristors is wide in range, like in-memory computing, logic, neuromorphic computing, etc.

There are several models of memristors [2]. While designing the circuits, a mathematical model is used to show the behaviour of the memristor [3], [4]. Compared to the behaviour of physical devices, the model should be sufficiently accurate, simple, and computationally efficient. In addition, the model should be general so that it can be suitable for different technologies. The wide usage of different memristor models for circuit simulations makes them flexible for a wide range of applications. While using the models [5] in large circuits for high-end applications, it is challenging to cross-validate.

Finding the efficient solution to several complex computational problems, evolving hardware neuromorphic computing architectures offer promising solutions. Memristors mimic synapses in neural network implementations, which change resistance state according to the applied voltage and memorise the latest attained resistance state. Like a matrix, the crossbar arrangement of memristors with selector transistors along rows and columns mimics weighted summation operations in neural network models. It offers different high-density architectures to implement the synaptic connections and neural network models. For this, the hardware circuit implementations based on different memristor models demand deciding the appropriate selection of these models to get maximum performance. In this proposed work, memristor models used in the circuit simulations within a black box are predicted using machine learning based on the dataset of pinched hysteresis.

This method has significant industrial applications in implementations of different neural network models, pattern recognition, in-memory computations, etc by enabling identification of proper memristor models suitable for specific computations that helps to optimize the neuromorphic systems. The tasks like image classification, language processing, speech recognition, robotics is flexible to be implemented with neuro memristive arrays with proper memristor models. The decision making process while using large datasets in different fields like finance, health care, manufacturing industries and edge computing applications using

TABLE 1. Properties and Challenges of Different Memristor Models

Memristor Models	Properties and Challenges
Joglekar and Biolek	There are a lot of discrepancies while comparing the results with the data obtained through physical characterization. When pulse-wave form is applied, in the positive regime, when conductivity increases, the size of the hysteresis loop increases, which is opposite to the trend seen in the characterization data.
Air Force Research Lab (AFRL) and metal-insulator-metal (MIM)	Correlate more closely to the characterization data, and there is a strong connection to the physical mechanisms within the device. For alternative voltage inputs, the results will not match the characterization data. Significant updation is needed with different device structures because these models are specific to a single fabricated device.
Hyperbolic Sine Models and the University of Michigan Model	The properties like the Schottky barrier at a metal-oxide interface and the diffusion of ions are modelled. Have a stronger correlation to physical memristor characterization data. These models describe the memristor functionalities in a more generalized and accurate way. However, they hardly correlate to the physical hardware.
Other generalized models	Less theoretical correlation to the physical mechanisms. Many models will not consider the threshold voltage of the physical memristor device. Unless the voltage across the memristor exceeds this threshold voltage, the hysteresis will not be seen. The state variable motion depends on the applied voltage's magnitude and polarity. This implies that the dynamics of Oxygen vacancy expansion and compression are different. Most models have the equivalent state variable motion in positive and negative directions, which is not true in actual cases.

smart sensors and IoT can use this neuromorphic computing architectures. Choosing proper memristor models that can have high performance to specific applications is essential for having higher degree of accuracy in computations. This proposed approach provides proper crossvaliation and prediction of memristor models so that the usage of those models with proper mapping to the application demands can be done.

This paper focuses on explaining neuro-memristive circuits and systems through a cross-validation of the simulations, irrespective of the complexity of the model. Memristors represent a broad class of devices that can be modelled using a broad set of device models. This problem is very different to that of MOSFETs, where only one type or minor variant of the device is modelled. Over time, even if models are standardized for memristors, there will be a need to perform cross-validations as being a class of devices, several combinations of variability make system modelling complex and inaccurate. The conventional system of verifying the simulation results of emerging memristive devices that are yet to mature, using the experimental results, is replaced here to address the challenges associated with the accurate modelling due to different variability. Even though many models are emerging, the scientific community always need to compromise for different properties associated with the physical devices like threshold voltage, state variable motion, area of hysteresis, different responses for different input stimuli, etc., due to the wide range of variability among this broad class of devices. This reverse engineering approach addresses the complexity of estimating the accuracy and functional behaviour of physical characterization data due to the above-mentioned issues. This proposed approach is helpful as more and more memristive devices are discovered.

A large majority of research using machine learning is for device modelling. However, this work diverges from this trend in applying machine learning in an entirely new application, where conventionally, cross-validation of simulation results is only done through experimental verification. This is an approach to cross-validate the simulations using a machine learning approach.

The motivation of this work is to develop a technique for explaining nero-memristive circuits and systems by validating circuit simulations done with emerging device models. Most memristor devices are difficult to model accurately due to device-to-device and cycle-to-cycle variability. Some examples of memristor models, their properties, device level and simulation level challenges are illustrated in table 1. Under such circumstances, the circuits built with idealistic models result in large output errors. Furthermore, as the devices have a range of variability, the experimental results are also difficult to conclude in estimating the desired functional behaviour and accuracy of the design logic. This necessitates a simulation-based approach to cross-validate the functionality and accuracy of circuit designs with memristors.

Through this new approach, we propose that estimating the device model followed by using those models to build circuits can lead to better estimates in cross-validating the accuracy of circuit-level simulation results. Reverse engineering the model from circuit design using the proposed approach also leads to an efficient way to account for a wide range of device variability.

The paper is organized as follows. The first section gives an introduction. The second section comprises the background of this paper. The third section details the proposed model prediction approaches in the single memristor model and models in memristive crossbar arrays, followed by the analysis methodology. The results and discussion are included in the fourth section, followed by the conclusion and references.

II. BACKGROUND

Most memristor models selected for this work are based on the memristor equations of HP memristor model.

A. HP MEMRISTOR ION-DRIFT MODEL

The principle of resistance switching between two extreme values, Ron and Roff, the device's lowest and highest resistance, makes them mathematically flexible to model in

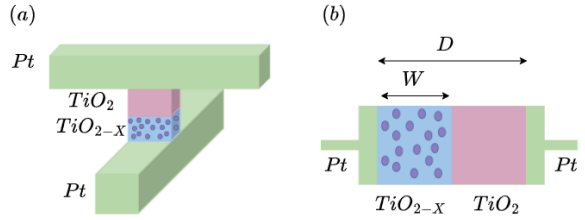

FIGURE 1. (a) HP ion drift model implementation illustrated in material physics. (b) The sandwich structure having two Pt electrodes, TiO_2 and TiO_{2-X}

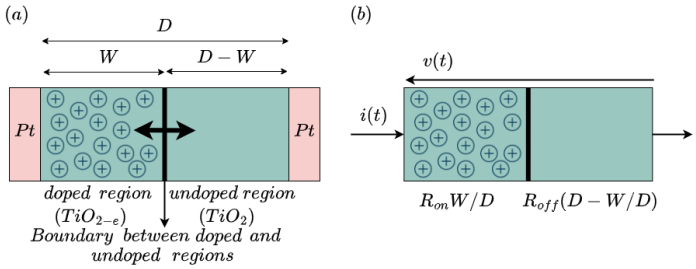

FIGURE 2. (a) The detailed device level structure of the HP memristor model. W is the width of the doped region, and D is the total device length. (b) The total resistance will be the effective resistance of the low resistance region (doped region) and the high resistance region (undoped region)

different ways. The HP memristor model, an example of a Metal-Insulator-Metal (MIM) device, is shown in figure 1.

When the Pt electrodes are excited externally, as shown in figure 2, the oxygen ions present in the doped region will drift to the undoped region under the influence of the electric field. This process will cause a shifting of the boundary between these two regions. This displacement in the boundary will cause a change in the resistance value also. If the structure is entirely covered by TiO_{2-x}, it is in its low resistance state or maximum conductive (R_{on}). If the structure is entirely covered by TiO_2, it is in its high resistance state (R_{off}) or minimum conductive. The Memristance is given by, $M(q) = R_{on}w(t)/D + R_{off}(1 - w(t)/D)$ The relation between the voltage and current is given by $V(t) = (R_{On}w(t)/D + R_{off}(1 - w(t)/D))i(t)$. Here w(t) is the width of the doped region, and D is the total width of the doped and undoped regions. The width w(t) is affected by i by, $dw/dt = \mu_v R_{on} i(t)/D$. Here μ_v is the dopant mobility. $w(t) = \mu_v R_{on} q(t)/D + w_0$. Here, q(t) is the charge injected in the time t.

The dw/dt is the dynamic state variable, the drift velocity of the Oxygen vacancies. The integration of the expression $\mu_v R_{on} q(t)/D$ gives the value of w(t). Even q(t)=0, the integrated output will equal a constant. This implies even if the current flow is zero, the charge is constant, and resistance remains unchanged. The principle of non-volatility satisfies here. Based on the migration of ions, the value of w varies

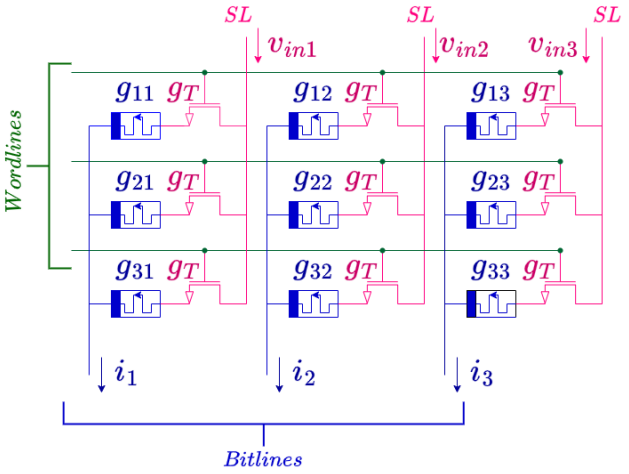

FIGURE 3. Memristor-crossbar architecture with inputs $v_{in1}, v_{in2}, v_{in3}$ and output currents i_1, i_2, i_3. Selector devices that need to be activated are applied with an input voltage greater than the threshold voltage. The Conductance of the memristor is denoted as g_{mn} for the m^{th} row and n^{th} column. The conductance of the selector transistor is given by g_T

between 0 and D. The drifting of the boundary region is interpreted by different window functions and equations that give different mathematical models of memristors. While modelling different memristors, state variable equations are substituted with the equation x(t)= W(t)/D. The state variable becomes a normalised quantity whose value lies between 0 and 1. x(t)=0 for the minor conductive state and x(t)=1 for the most conductive state. Window functions limit the motion of the state variable between 0 and 1.

B. MEMRISTIVE CROSSBAR ARRAY

In a crossbar architecture [6], [7], memristors are arranged in a matrix form, as shown in figure 3. Each row and column intersection consists of a memristive device [8]. Word lines in a crossbar feed input voltage, and bit lines are used to read output currents. The output currents are the results of the Multiply and Accumulate(MAC) operation [9] between the input voltage and conductance of the memristive device. Each memristor device is accompanied by a selector device to select the desired device among the rows and columns.

C. ENSEMBLE LEARNING - RANDOM FOREST AND EXTREME GRADIENT BOOSTING

Ensemble learning techniques combine different learning algorithms to make more accurate predictions. Predictions from individual learning models are aggregated to form the final prediction. These algorithms efficiently handle non-linearity and interactions and provide feature-importance, flexibility and robustness to overfitting. Since the data collected for this study is susceptible to overfitting the model and shows a non-linear relationship, we are focusing on the following two ensemble learning techniques.

Random forest [10], [11] is an ensemble prediction algorithm having a combination of tree predictors. The majority

vote of all individual trees determines the final prediction of the input vector. Each tree casts one vote for the most frequently occurring class. In Random Forest, the Gini Index, which measures the degree of impurity of an attribute to different classes, is used as an attribute selection measure.

For a given training set T, selecting one case randomly and assigning to a class C_i, the Gini index is expressed by:

$$\sum \sum_{j \neq i} (f(C_i, T)/|T|)(f(C_j, T)/|T|) \quad (1)$$

Where $f(C_i, T)/|T|$ is the probability that the selected case belongs to class C_i. To reduce the complexity of the tree and to prevent overfitting, pruning is used [12]. It removes the branches of the tree that do not contribute to accuracy, and the remaining branches are grown to the maximum.

In this approach, input voltages and the corresponding output currents are the features. Based on this data, the Random forest prediction algorithm splits nodes to generate new trees and identifies the respective models.

The Extreme Gradient Boosting (XGBoost) [13] is a scalable and efficient application of the gradient boosting framework. A weight will be assigned for each observation. This weight will be adjusted after training the predictor. The weight of the correctly classified observations is decreased, and misclassified observations are increased. Using the observations with modified weights, the subsequent predictor is trained, and the process is repeated to create a highly accurate model. The sum of prediction score $f_k(X_i)$ of all trees gives the estimated output $\hat{y_i}$of the gradient boosting tree model

$$\hat{y_i} = \sum_{k=1}^{K} f_k(X_i), f_k \in \Gamma \quad (2)$$

where Γ: Space of the regression tree, K: The number of regression trees, X_i: The features corresponding to sample I.

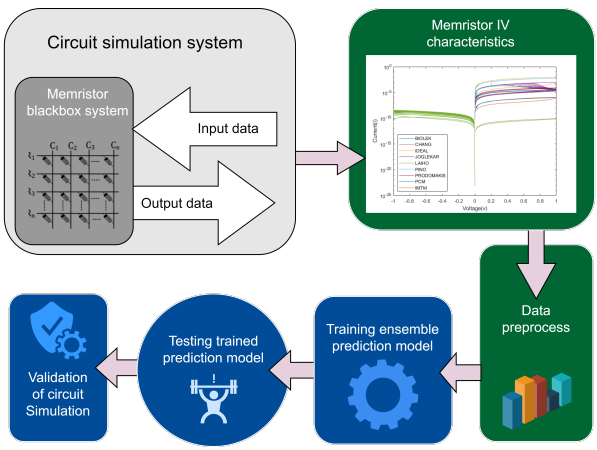

FIGURE 4. Explainable neuro-memristive circuit system workflow

III. PROPOSED MEMRISTOR MODEL PREDICTOR

This approach proposes estimating the device model and using those models to build circuits to get better estimates in cross-validating the accuracy of circuit-level simulation results. Reverse engineering the model from circuit design also leads to an efficient way to account for a wide range of device variability. A workflow of the proposed approach is shown in the figure 4.

Here, nine memristor models are simulated in Spice. Each model is simulated with an input voltage of four different frequencies (0.5Hz, 1Hz, 5Hz and 10 Hz). The graph obtained by plotting the input voltage versus the logarithmic scale of output current gives nine different pinched hystereses, as shown in figure 5. This output data is collected for each model and performed prediction using the Random forest and XGBoost techniques. For random forest, the parameters are trained with 100 trees for each data set for the different frequencies applied. The prediction results for these nine models with four different frequencies of input voltages using the Random forest and XGBoost algorithms are analysed by the factors precision, recall, f1-score and support. The two prediction approaches, accuracy, macro average and weighted average, are compared for different frequencies.

To calculate precision, recall, and f1-score, the following parameters are calculated from the confusion matrix [14], [15]. A confusion matrix is a tabular way of representing the performance of the prediction algorithm

True positive (TP): Values predicted as positive and it's true. True negative (TN): Values predicted as negative and it is true. False positive (FP): Values predicted as positive and it is false. False negative (FN): Values predicted as negative and it is false.

Precision is found using the following equation

$$Precision = TP/(TP + FP)$$

Recall is calculated by the following equation

$$Recall = TP/(TP + FN)$$

F1-score is measured using the equation

$$F1{-}score = (2{*}Recall{*}Precision)/(Recall{+}Precision)$$

For performing prediction in a memristive crossbar instead of a single memristor, memristors are arranged row and column-wise in different dimensions (2×2, 4×4, 8×8 and 16×16). The simulated spice data of different input voltages through rows and different output currents through the columns are used to predict using Random forest and XGBoost. Performance is analysed based on the factors of precision, recall, f1-score and support. The accuracy, macro average and weighted average of the two prediction approaches are compared for different crossbar dimensions.

A. RANDOM FOREST ALGORITHM ON MEMRISTOR DATA

A random forest algorithm is used for prediction and regression problems. It combines multiple decision trees to form a forest. To predict, a random subset of the input data and a random subset of the input features are used to train each

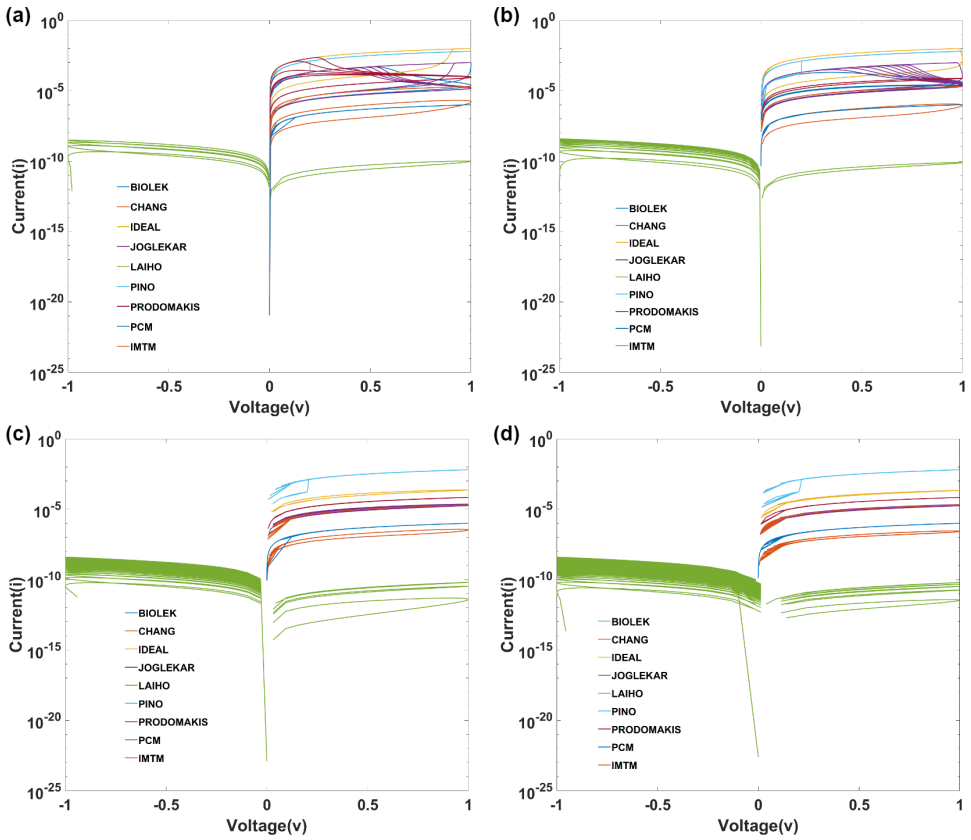

FIGURE 5. BIOLEK: Biolek Model, UMM: University of Michigan Model, IDEAL: Ideal Memristor Model, JOGLEKAR: Joglekar Model, GHSM: General Hyperbolic Sine Model, AFRLM: Air Force Research Lab Model, PRODOMAKIS: Prodromakis Model, PCM: Phase Change Memory, IMTMS: Insulator to Metal Transition Memristive Systems, Pinched hysteresis of nine memristor models with a) 0.5Hz input voltage frequency b) 1Hz input voltage frequency c) 5Hz input voltage frequency and d) 10Hz input voltage frequency

TABLE 2. Performance Analysis of Random Forest predictor on single memristor circuit simulations

Memristor Model	0.5 Hz				1 Hz				5 Hz				10 Hz			
	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
BIOLEK	0.75	0.80	0.77	426	0.92	0.84	0.88	427	0.71	0.86	0.78	1306	0.85	0.84	0.85	1366
UMM	0.66	0.94	0.78	298	0.73	0.89	0.80	462	0.72	1.00	0.83	1423	0.94	0.81	0.87	1348
IDEAL	0.93	0.93	0.93	338	0.97	0.99	0.98	633	1.00	1.00	1.00	950	1.00	1.00	1.00	1393
IMTMS	0.88	0.89	0.89	170	0.93	0.95	0.94	321	1.00	0.97	0.98	1123	0.96	0.99	0.98	2271
JOGLEKAR	0.78	0.74	0.76	565	0.93	0.91	0.92	932	0.82	0.66	0.73	1342	0.86	0.84	0.85	1302
GHSM	0.93	0.76	0.84	214	0.97	0.90	0.93	404	1.00	0.86	0.93	1236	0.84	0.93	0.88	1348
PCM	0.71	0.40	0.51	138	0.72	0.66	0.69	456	1.00	0.51	0.68	756	0.91	0.93	0.92	1380
AFRLM	0.98	0.98	0.98	1227	1.00	0.99	0.99	2509	1.00	1.00	1.00	13105	1.00	1.00	1.00	28767
PRODOMAKIS	0.88	0.82	0.85	439	0.97	0.98	0.97	431	1.00	1.00	1.00	1382	1.00	1.00	1.00	1377

decision tree. Aggregation of the decisions of all trees gives the final decision of prediction. Here, we use a technique known as bagging that reduces overfitting and improves accuracy by combining the predictions of multiple decision trees formed from bootstrapped training data samples. The Pseudocode for Random Forest is given in Algorithm 1.

B. XGBOOST ALGORITHM ON MEMRISTOR DATA

XGBoost algorithm is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. Decision trees are base learners for the XGBoost or eXtream Gradient Boosting prediction. It controls overfitting by using a more regularised model. This makes it more accurate and faster than traditional gradient boosting. The Pseudocode for XGBoost is given in Algorithm 2.

IV. RESULTS AND DISCUSSION

The nine memristor models with four different frequencies used for the prediction are shown in figure 5. The predictor may not capture the relevant information if the number of features is too small. If the number of features is too large, the predictor may overfit the training data, leading to poor generalisation performance. Since we are only using two features, input voltage and output current, the predictor highly depends on the data. In the dataset, different models show similar readings of input voltage and output current (at the pinched point). Here, the overall data is set into 70% of training data

TABLE 3. Performance Analysis of XGBoost predictor on single memristor circuit simulations

Memristor Model	0.5 Hz				1 Hz				5 Hz				10 Hz			
	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support	precision	recall	f1-score	support
BIOLEK	0.72	0.71	0.71	431	0.86	0.74	0.80	415	0.61	0.87	0.72	1334	0.87	0.94	0.91	1349
UMM	0.91	0.98	0.94	301	0.98	0.86	0.91	418	1.00	1.00	1.00	1437	0.99	1.00	1.00	1335
IDEAL	0.91	0.86	0.88	380	0.95	0.97	0.96	626	1.00	1.00	1.00	913	1.00	1.00	1.00	1360
IMTMS	0.69	0.90	0.78	165	0.76	0.90	0.82	339	0.97	0.76	0.85	1069	0.95	0.97	0.96	2245
JOGLEKAR	0.75	0.73	0.74	565	0.85	0.81	0.83	904	0.75	0.56	0.64	1354	0.95	0.84	0.89	1380
GHSM	0.99	0.99	0.99	190	1.00	1.00	1.00	412	1.00	1.00	1.00	1198	1.00	1.00	1.00	1312
PCM	0.95	0.80	0.87	148	0.87	0.98	0.92	467	1.00	0.99	0.99	724	1.00	0.99	1.00	1348
AFRLM	0.98	0.96	0.97	1223	0.99	0.98	0.98	2603	1.00	1.00	1.00	13308	1.00	1.00	1.00	28852
PRODOMAKIS	0.76	0.78	0.77	412	0.90	0.97	0.93	391	0.99	1.00	1.00	1286	1.00	1.00	1.00	1371

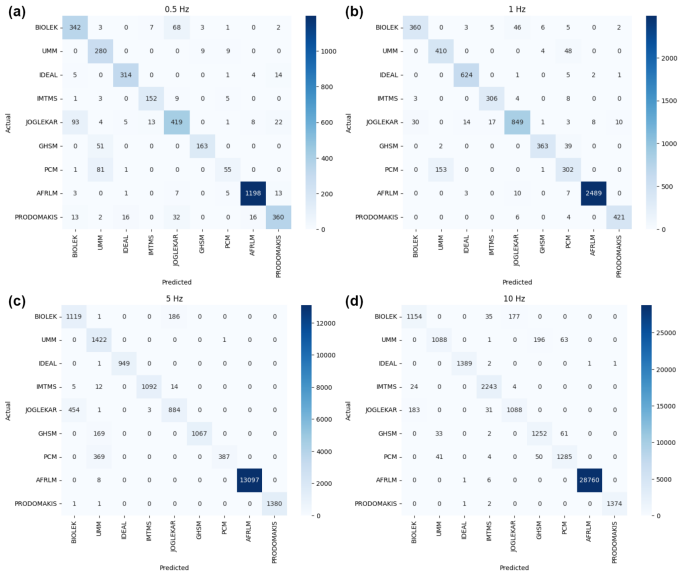

FIGURE 6. Random Forest prediction confusion matrix for single memristor simulation for a) 0.5 Hz input voltage frequency, b) 1 Hz input voltage frequency, c) 5 Hz input voltage frequency and d) 10 Hz input voltage frequency

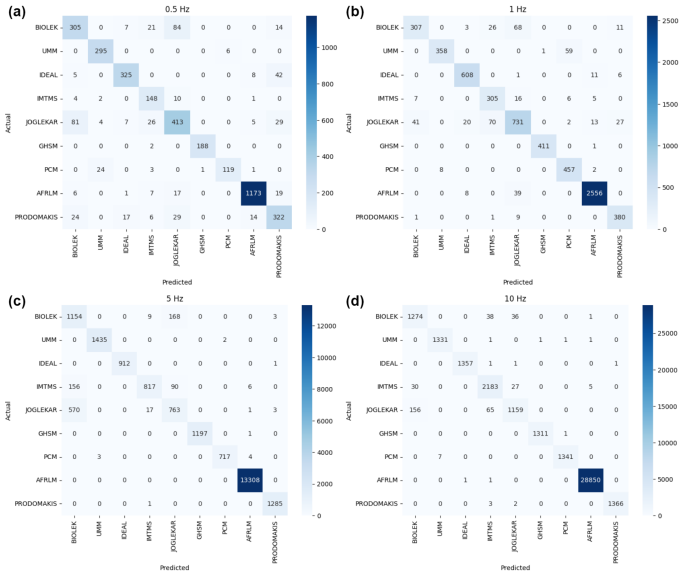

FIGURE 7. XGBoost prediction confusion matrix for single memristor simulation for a) 0.5 Hz input voltage frequency, b) 1 Hz input voltage frequency, c) 5 Hz input voltage frequency and d) 10 Hz input voltage frequency

and 30% of test data.

The initial dataset is split for training and testing. The testing dataset contains randomly selected data for each model. After training to predict the model, this testing dataset is used as input. After testing both algorithms in various data, the performance is visualized using a confusion matrix and analyzed using the parameters precision, recall, and f1-score based on the support for each model. Precision is the ratio of the number of true positives to the number of elements labelled to belong to the positive class. The ratio between the number of true positives and the total number of elements that belongs to the positive class gives recall, and f1-score is calculated by taking the harmonic mean of precision and recall. Support represents the number of samples of true responses lying in the class. The overall performance is evaluated using accuracy, macro average, and weighted average. In macro average, all classes equally contribute to the final averaged matrix, and in weighted average, each class's contribution to the average is weighted by its size.

The random forest prediction technique in which the parameters are trained with 100 trees is used to perform prediction. table 2 and table 3 shows the performance analysis of the Random forest and XGBoost predictors for these nine models

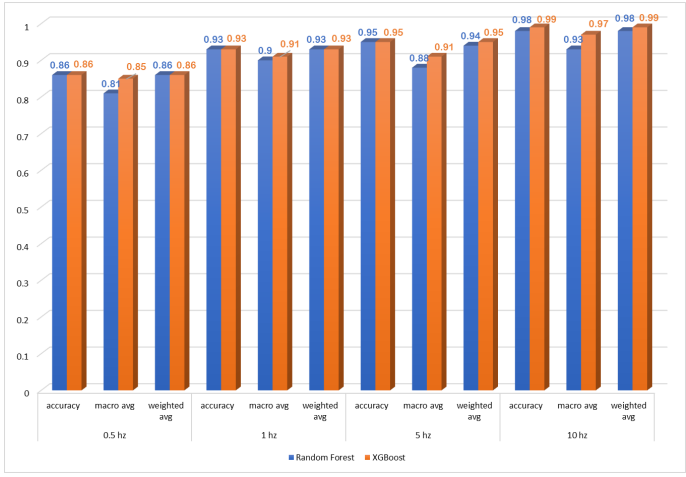

FIGURE 8. Accuracy, macro average and weighted average while using Random forest and XGBoost for single memristor models at frequencies 0.5Hz, 1Hz, 5Hz and 10Hz

Algorithm 1 Pseudocode for Random Forest Classifier

Training on the data

Training data ($X_{\text{train}}, y_{\text{train}}$), Number of trees (`num_trees`), Max depth of each tree (`max_depth`) Random Forest model

Procedure:

Initialize an empty list to store the trees: `forest = []`

for $i \leftarrow 1$ `num_trees` **do** `subset_X, subset_y = random_subset(`$X_{\text{train}}, y_{\text{train}}$`)`

`tree = build_decision_tree(subset_X, subset_y, max_depth)`

`forest.append(tree)`

Return: Random Forest model (`forest`)

Prediction using the trained model

Random Forest model, Test data (`X_test`) Predicted class labels for `X_test`

Procedure:

Initialize an array to store the predictions: `predictions = []`

for each `tree` in `forest` **do** `prediction = predict_with_tree(tree, X_test)`

`predictions.append(prediction)`

Return: Majority vote of `predictions`

Algorithm 2 Pseudocode for XGBoost Classifier

Training on the data

Training data ($X_{\text{train}}, y_{\text{train}}$), Number of boosting rounds (`num_rounds`), Maximum depth of each tree (`max_depth`), Learning rate (`eta`), Subsample ratio of training instances (`subsample`), Column subsample ratio of features (`colsample_bytree`) XGBoost model

Procedure:

Initialize model with a constant value: `model = initial_prediction_value`

for round $\leftarrow 1$ `num_rounds` **do**

`gradients = -gradient_of_loss(`y_{train}`, model.predict(`X_{train}`))`

`weak_model = fit_weak_model(`X_{train}`, gradients, max_depth, colsample_bytree)`

`update = eta * weak_model.predict(`X_{train}`)`

`model = model + update`

Return: XGBoost model (`model`)

Prediction using the trained model

XGBoost model, Test data (`X_test`) Predicted class labels for `X_test`

Procedure:

`predictions = model.predict(`X_{test}`)`

Return: `predictions`

with four different frequencies of input voltages analysed by the factors precision, recall, f1-score and support using the confusion matrix shown in figures 6 and 7 respectively. Most of the models give high accuracy while using both prediction techniques. Support is a significant factor considering the prediction parameters for individual memristor models. Depending on the support, both algorithms show varying performance parameters but are still well enough to identify the model successfully. The comparison of overall accuracy, macro average and weighed average of Random forest and XGBoost for single memristor models at four different frequencies 0.5 Hz, 1 Hz, 5 Hz and 10 Hz are illustrated by the figure 8. The performance of both approaches enhances with frequency. Maximum accuracy is achieved at a higher frequency. This implies that even though there are similar points in the dataset, the two predictors can produce a better accuracy in predicting or identifying the model. The performance analysis of Random forest prediction and XGBoost prediction for 2×2 crossbar, 4×4 crossbar, 8×8 crossbar and 16×16 crossbar based on the confusion matrix shown in the figures 9 and 10 are illustrated in the tables 4, 5, 6 and 7 respectively. According to the performance parameters, support plays a significant role in predicting the model. Since the data from each model are close enough for higher prediction accuracy, more input data points are required for higher accuracy. Accuracy, macro average and weighted average are compared for the two prediction approaches for the above four dimensions, as shown in figure 11. Random forest and XGBoost gave more than 80% overall accuracy in four cases. In some cases, Random forest performed better than the XGBoost algorithm. But this can vary depending on the input data. Both approaches perform well enough to predict the model in most cases.

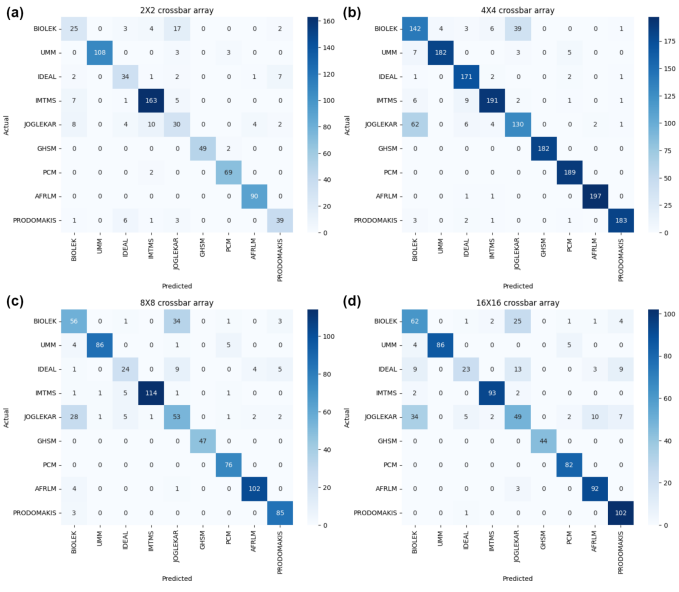

FIGURE 9. Random Forest prediction confusion matrix for memristor crossbar array simulation of a) 2X2 crossbar array, b) 4X4 crossbar array simulation, c) 8X8 crossbar array and d) 16X16 crossbar array

V. CONCLUSION

The proposed work classifies and predicts the memristor models used in circuit simulations with only available data

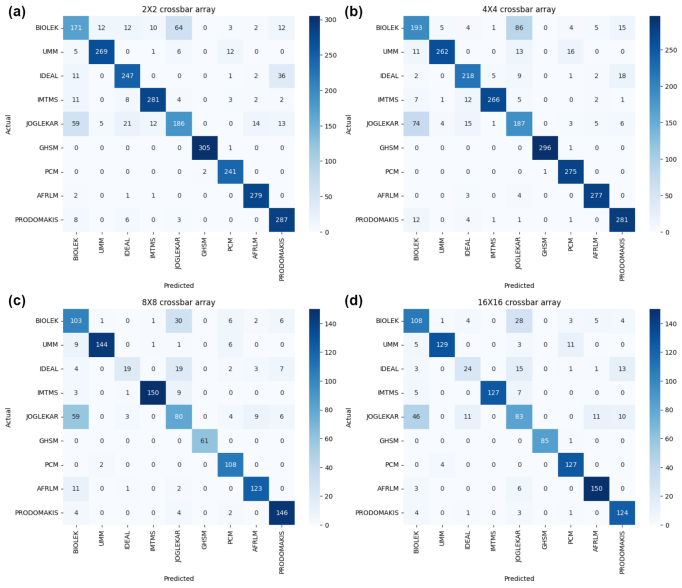

FIGURE 10. XGBoost prediction confusion matrix for memristor crossbar array simulation of a) 2X2 crossbar array b) 4X4 crossbar array c) 8X8 crossbar array and d) 16X16 crossbar array simulation

TABLE 4. Performance Analysis of Random Forest and XGBoost algorithm on 2×2 Crossbar

Memristor Model	Random Forest				XGBoost			
	precision	recall	f1-score	support	precision	recall	f1-score	support
BIOLEK	0.58	0.49	0.53	51	0.64	0.60	0.62	286
UMM	1.00	0.95	0.97	114	0.94	0.92	0.93	293
IDEAL	0.71	0.72	0.72	47	0.84	0.83	0.83	297
IMTMS	0.90	0.93	0.91	176	0.92	0.90	0.91	311
JOGLEKAR	0.50	0.52	0.51	58	0.71	0.60	0.65	310
GHSM	1.00	0.96	0.98	51	0.99	1.00	1.00	306
PCM	0.93	0.97	0.95	71	0.92	0.99	0.96	243
AFRLM	0.95	1.00	0.97	90	0.93	0.99	0.96	283
PRODOMAKIS	0.78	0.78	0.78	50	0.82	0.94	0.88	304

of inputs and outputs. The efficient hardware neuromorphic computing systems for different industrial applications can be implemented with higher degree of performance by choosing memristor models that suit specific applications. Exploring other advanced classifiers for prediction and cross-validation can enlarge the boundaries for industrial applications in which this explainable AI approach can be used. In this paper, we propose two ensemble learning techniques, Random Forest and XGBoost, to cross-validate circuit simulations for different models of Memristors. These proposed prediction models estimate the memristor model from a circuit simulation's voltage and current measurements. In the

TABLE 5. Performance Analysis of Random Forest and XGBoost algorithm on 4×4 Crossbar

Memristor Model	Random Forest				XGBoost			
	precision	recall	f1-score	support	precision	recall	f1-score	support
BIOLEK	0.64	0.76	0.69	193	0.65	0.62	0.63	313
UMM	0.98	0.95	0.97	195	0.96	0.87	0.91	302
IDEAL	0.88	0.97	0.92	150	0.85	0.85	0.85	255
IMTMS	0.98	0.90	0.94	225	0.97	0.90	0.94	294
JOGLEKAR	0.76	0.62	0.68	219	0.61	0.63	0.62	295
GHSM	0.99	1.00	1.00	196	1.00	1.00	1.00	297
PCM	0.98	1.00	0.99	189	0.91	1.00	0.95	276
AFRLM	0.97	0.99	0.98	196	0.95	0.98	0.96	284
PRODOMAKIS	0.95	0.98	0.96	181	0.88	0.94	0.90	300

TABLE 6. Performance Analysis of Random Forest and XGBoost algorithm on 8×8 Crossbar

Memristor Model	Random Forest				XGBoost			
	precision	recall	f1-score	support	precision	recall	f1-score	support
BIOLEK	0.58	0.59	0.58	95	0.53	0.69	0.60	149
UMM	0.98	0.90	0.93	96	0.98	0.89	0.94	161
IDEAL	0.69	0.56	0.62	43	0.79	0.35	0.49	54
IMTMS	0.99	0.93	0.96	123	0.99	0.92	0.95	163
JOGLEKAR	0.54	0.57	0.55	93	0.55	0.50	0.52	161
GHSM	1.00	1.00	1.00	47	1.00	1.00	1.00	61
PCM	0.90	1.00	0.95	76	0.84	0.98	0.91	110
AFRLM	0.94	0.95	0.95	107	0.90	0.90	0.90	137
PRODOMAKIS	0.89	0.97	0.93	88	0.88	0.94	0.91	156

TABLE 7. Performance Analysis of Random Forest and XGBoost algorithm on 16×16 Crossbar

2*Memristor Model	Random Forest				XGBoost			
	precision	recall	f1-score	support	precision	recall	f1-score	support
BIOLEK	0.56	0.65	0.60	96	0.62	0.71	0.66	153
UMM	1.00	0.91	0.95	95	0.96	0.87	0.91	148
IDEAL	0.77	0.40	0.53	57	0.60	0.42	0.49	57
IMTMS	0.96	0.96	0.96	97	1.00	0.91	0.95	139
JOGLEKAR	0.53	0.45	0.49	109	0.57	0.52	0.54	161
GHSM	1.00	1.00	1.00	44	1.00	0.99	0.99	86
PCM	0.91	1.00	0.95	82	0.88	0.97	0.92	131
AFRLM	0.87	0.97	0.92	95	0.90	0.94	0.92	159
PRODOMAKIS	0.84	0.99	0.91	103	0.82	0.93	0.87	133

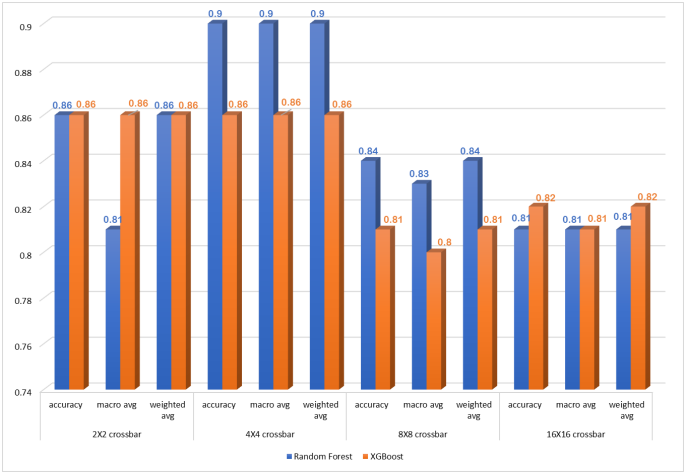

FIGURE 11. Accuracy, macro average, weighted average while using Random forest and XGBoost for crossbar architectures of dimensions 2×2,4×4,8×8 and 16×16

detailed examinations, we found that the predictive models could perform with high accuracy in various configurations of the circuit design simulations. The final analysis is based on the accuracy, precision and f1-score obtained from the confusion matrix. The input voltage frequency was a key component in the accuracy of the prediction models. The prediction model's accuracy increased with frequency. In various crossbar simulations, the prediction models performed with high accuracy. In Some cases, Random Forest was able to perform better than XGBoost. From the final analysis, we concluded that Random Forest and XGBoost work well given large homogeneous training data and are relatively robust to outliers.

The proposed prediction methods cross-validate the memristor circuit simulations, ensuring accurate results concerning the memristor model. This method helps precisely anal-

yse the circuit's IV characteristics and reverse engineering the circuit only from the output measurements. When there is confusion on which memristor model should be used for a desired input and output, this cross-validation system can be successfully implemented to explain the black-box mystery.

REFERENCES

[1] L. Chua, "Memristor-the missing circuit element," IEEE Transactions on circuit theory, vol. 18, no. 5, pp. 507–519, 1971.

[2] Z. Biolek, D. Biolek, and V. Biolkova, "Spice model of memristor with nonlinear dopant drift." Radioengineering, vol. 18, no. 2, 2009.

[3] P. Sheridan and W. Lu, "Memristors and memristive devices for neuromorphic computing," in Memristor Networks, pp. 129–149. Springer, 2014.

[4] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," Nature nanotechnology, vol. 8, no. 1, pp. 13–24, 2013.

[5] S. Kvatinsky, K. Talisveyberg, D. Fliter, A. Kolodny, U. C. Weiser, and E. G. Friedman, "Models of memristors for spice simulations," in 2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel, DOI 10.1109/EEEI.2012.6377081, pp. 1–5, 2012.

[6] A. P. James and L. O. Chua, "Analog neural computing with super-resolution memristor crossbars," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 68, no. 11, pp. 4470–4481, 2021.

[7] I. Vourkas, D. Stathis, G. C. Sirakoulis, and S. Hamdioui, "Alternative architectures toward reliable memristive crossbar memories," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 1, pp. 206–217, 2015.

[8] T. Prodromakis and C. Toumazou, "A review on memristive devices and applications," in 2010 17th IEEE international conference on electronics, circuits and systems, pp. 934–937. IEEE, 2010.

[9] J. Chen, J. Li, Y. Li, and X. Miao, "Multiply accumulate operations in memristor crossbar arrays for analog computing," Journal of Semiconductors, vol. 42, no. 1, p. 013104, 2021.

[10] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," ISPRS journal of photogrammetry and remote sensing, vol. 67, pp. 93–104, 2012.

[11] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How many trees in a random forest?" in Machine Learning and Data Mining in Pattern Recognition: 8th International Conference, MLDM 2012, Berlin, Germany, July 13-20, 2012. Proceedings 8, pp. 154–168. Springer, 2012.

[12] V. Y. Kulkarni and P. K. Sinha, "Pruning of random forest classifiers: A survey and future directions," in 2012 International Conference on Data Science & Engineering (ICDSE), pp. 64–68. IEEE, 2012.

[13] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," Annals of statistics, pp. 1189–1232, 2001.

[14] S. Haghighi, M. Jasemi, S. Hessabi, and A. Zolanvari, "Pycm: Multiclass confusion matrix library in python," Journal of Open Source Software, vol. 3, no. 25, p. 729, 2018.

[15] M. Heydarian, T. E. Doyle, and R. Samavi, "Mlcm: Multi-label confusion matrix," IEEE Access, vol. 10, pp. 19 083–19 095, 2022.

RAHUL KOTTAPPUZHACKAL is a research assistant at the School of Electronics Systems and Automation, Digital University of Kerala. He is also working as an AI engineer at India Graphene Engineering and Innovation Centre. Rahul received a Master's in Computer Science in 2023 and a bachelor's in Physics in 2020. He currently involved in the areas of machine learning, memristive systems, and neuromorphic computing systems. His research interests include the applications of machine learning for memristor-based circuits.

SRUTHI PALLATHUVALAPPIL is a PhD student at the School of Electronics Systems and Automation, Digital University of Kerala. Sruthi's research area focuses on low-power resistive memory networks for AI. Sruthi received a Master of Technology in Embedded Systems in 2017 and Bachelors degree in Electronics and Communication in 2014. Sruthi is currently involved in a few projects related to hardware-based low power memristive network implementation. Sruthi's areas of interest include memristive analog circuits, multi-bit logic memories, 3D integration, and neuromorphic computing systems. She is a graduate student member of IEEE.

ALEX JAMES received the Ph.D. degree from Griffith University, Queensland, Australia. He is currently a Professor and the Dean (Academic) with the Kerala University of Digital Sciences, Innovation and Technology (aka Digital University Kerala). He is the Professor-in-Charge of the Maker Village, Chief Investigator with the India Innovation Centre for Graphene, Head of AI Chip Centre, and Chief Scientist/CTO for India Graphene Engineering and Innovation Centre. He is also advisory board member of Digital Science Park. His research interests include AI - neuromorphic systems (software and hardware), VLSI and image processing. He is a Member of IEEE CASS TC on Nonlinear Circuits and Systems, IEEE CTSoc TC on Quantum in Consumer Technology (QCT), TC on Machine learning, Deep learning and AI in CE (MDA), IEEE CASS TC on Cellular Nanoscale Networks and Memristor Array Computing (CNN-MAC), and IEEE CASS SIG on AgriElectronics. He was the founding Chair of IEEE CASS Kerala chapter, a Member of IET Vision and Imaging Network, and currently a Member of BCS' Fellows Technical Advisory Group (F-TAG). He was an Editorial Board Member of Information Fusion (2010–2014), and currently serving as an Associate Editor of IEEE Access, Frontiers in Neuroscience, IEEE Transactions on Circuits and System 1: Regular Papers and IEEE Open Journal of Circuits and Systems journal. He is a Life Member of ACM, Senior Fellow of HEA, Fellow of RSA, Fellow of British Computer Society (FBCS), and Fellow of IET (FIET).