


# Electric theft detection in advanced metering infrastructure using Jaya optimized combined Kernel-Tree boosting classifier—A novel sequentially executed supervised machine learning approach

Saddam Hussain<sup>1</sup>  | Mohd. Wazir Mustafa<sup>1</sup> | Khalil Hamdi Ateyeh Al-Shqeerat<sup>2</sup> |  
Bander Ali Saleh Al-rimy<sup>3</sup> | Faisal Saeed<sup>4</sup>

<sup>1</sup> School of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru, Malaysia

<sup>2</sup> Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

<sup>3</sup> School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia

<sup>4</sup> School of Computing and Digital Technology, Birmingham City University, Birmingham, UK

## Correspondence

Bander Ali Saleh Al-rimy, School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor 81310, Malaysia.

Email: [bander@utm.my](mailto:bander@utm.my)

Saddam Hussain, School of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia.

Email: [hussainsaddam@graduate.utm.my](mailto:hussainsaddam@graduate.utm.my)

## Funding information

Fundamental Research Grant Scheme, Grant/Award Number: R.J130000.7851.5F062; Ministry of Higher Education, Malaysia

## Abstract

This paper presents a novel, sequentially executed supervised machine learning-based electric theft detection framework using a Jaya-optimized combined Kernel and Tree Boosting (KTBoost) classifier. It utilizes the intelligence of the XGBoost algorithm to estimate the missing values in the acquired dataset during the data pre-processing phase. An oversampling algorithm based on the Robust-SMOTE technique is utilized to avoid the unbalanced data class distribution issue. Afterward, with the aid of few very significant statistical, temporal, and spectral features extracted from the acquired kWh dataset, the complex underlying data patterns are comprehended to enhance the accuracy and detection rate of the classifier. For effectively classifying the consumers into “Honest” and “Fraudster,” the ensemble machine learning-based classifier KTBoost, with Jaya algorithm optimized hyperparameters, is utilized. Finally, the developed model is re-trained using a reduced set of highly important features to minimize the computational resources without compromising the performance of the developed model. The outcome of this study reveals that the proposed theft detection method achieves the highest accuracy (93.38%), precision (95%), and recall (93.18%) among all the studied methods, thus signifying its importance in the studied area of research.

## 1 | INTRODUCTION

The integration of communication and information technologies with electrical infrastructure has become more prevalent in recent years. Smart grids, the next generation of energy distribution networks, are emerging due to the increasing penetration of advances in modern technology [1, 2]. One of the crucial components of smart grids is Advanced Metering Infrastructure (AMI) which allows the transfer of two-way data like time and quantity of energy used by a customer. With this new bi-directional information flow, AMI facilities power companies to perform accurate modelling of the cus-

tomers energy consumption behaviour [3], including predicting energy usage [4], demand response [5], and real-time pricing [6]. However, despite numerous advantages, threats such as cyber-attacks, smart meter hacking, and malicious data manipulation restrict the vast expansion of AMI [7–9] and jeopardize the grid's security. The most significant consequence of AMI is Non-Technical Losses (NTL) which accounts for power theft, errors in the metering/registering process, and invoicing mistakes [10]. Among all the mentioned NTL causes, electric power theft shares the major portion. Theft of power is not only associated with economic loss, but it also affects the power quality, increased load on the generating stations, and irrational

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *IET Generation, Transmission & Distribution* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology

tariffs imposed on legitimate consumers. Power utilities all over the globe incur significant revenue loss as a result of power theft. In the United States alone, this loss ranges from 0.5 percent to 3.5 percent of their annual income [11]. The case is even worse in underdeveloped nations where the revenue loss from this type of NTL becomes a significant portion of their gross domestic product [12, 13].

To decrease the NTLs, power utilities check all suspected consumers daily or weekly and then enforce punitive measures for any proven fraud practices. However, this process is time-consuming, expensive, and error prone. Currently, the majority of the power utilities, especially in under-developed countries, are employing traditional inefficient, laborious, costly, and time-consuming NTL detection systems. Nevertheless, in recent years, a significant increase in the deployment of AMI in distribution networks has been witnessed, which provides additional features such as monitoring, storing, and retrieving a broad variety of data at any time. In addition, data-oriented algorithms have been developed as an effective automated tool for screening aberrant energy consumption patterns and identifying possible electrical fraud activities. These data-oriented theft detection methods can be broadly categorized into four categories, statistical-based [14–17], game-theory-based [18, 19], expert system [20, 21] and ML-based [22–25].

## 1.1 | Major and minor contributions of the proposed theft detection system

This study endeavours to develop a novel supervised machine learning (SML)-based sequentially executed electricity theft detection framework that effectively detects fraudster consumers from an acquired smart meter dataset. The simplified flowchart of the developed method is illustrated in Figure 1, and the brief explanation of each executed novel stage is as follows

- (i) The proposed framework initiates its operation by substituting the missing entries in the obtained smart meter dataset using the machine learning (ML)-based predictive modelling technique. This technique estimates the missing data records by employing the XGBoost algorithm in such a manner that missing attributes act as the target class and the rest of the feature set as an input for model training. The important aspects of this algorithm include handling various kinds of missing data, being adaptable to interactions and non-linearity within the dataset, and being scalable to large data situations.
- (ii) After handling the missing values problem, the data class imbalance issue is addressed by using the robust synthetic minority oversampling approach (robust-SMOTE). The robust-SMOTE technique generates the minority samples (i.e., fraud cases) from all minority sample regions present in the dataset, such as those which are present within the majority class area (Healthy cases), on the borderline of the majority class, and the one which is far away

from the majority class samples. Subsequently, to accurately depict the underlying properties of consumption data, the proposed method utilizes the statistical, temporal, and spectral domains to extract features from collected consumption data.

- (iii) After collecting the most relevant characteristics, the model training-testing procedure is commenced by classifying customers into two different groups (“Genuine/Healthy” and “Theft/Fraudster”) using the KTBoost algorithm. The KTBoost algorithm combines kernel boosting and tree boosting methods for classification purposes. In each boosting iteration, it either adds a regression tree or a penalized reproducing kernel Hilbert space RKHS/kernel ridge regression function to the ensemble of base classifiers. Later, to obtain the best possible results, the model’s hyperparameters are tuned using a meta-heuristic-based optimization technique called the Jaya algorithm. The Jaya algorithm is a stochastic population-based optimization technique that modifies a population of individual solutions on an ordered basis by keeping the notion that each individual solution strives to attain the best solution while avoiding the least fit/worst one.
- (iv) Finally, the proposed model is retained with a smaller set of highly significant features while maintaining the same degree of accuracy, thus conserving computing resources.

In Section-2, the most relevant literature on the challenges encountered during the development of the SML framework is discussed. Section 3 discusses data exploration, the missing values imputation approach, the data class balancing method, feature engineering, and the theoretical background of the KTBoost and Jaya algorithms. Section 4 provides the outcomes of the proposed research work. Finally, Section 5 of this study contains the conclusion.

## 2 | LITERATURE REVIEW

The current research explores an application of the supervised ML-based theft detection framework; therefore, the most relevant information and literature are highlighted to understand better the proposed methodology and its significance in the studied field of research.

Typically, SML-based NTL detection techniques encounter five major issues:

- a. Handling of missing and outlying values occurrence in the accumulated raw dataset
- b. Target/data class imbalance distribution
- c. Method for relevant features extraction and selection
- d. The right choice of classification algorithm and its hyperparameters to maximize the prediction accuracy
- e. Understanding/interpreting the model’s prediction.

A number of attempts have been made in the literature to solve these issues, out of which few prominent research works are cited as per the sequence of the above-mentioned problems.

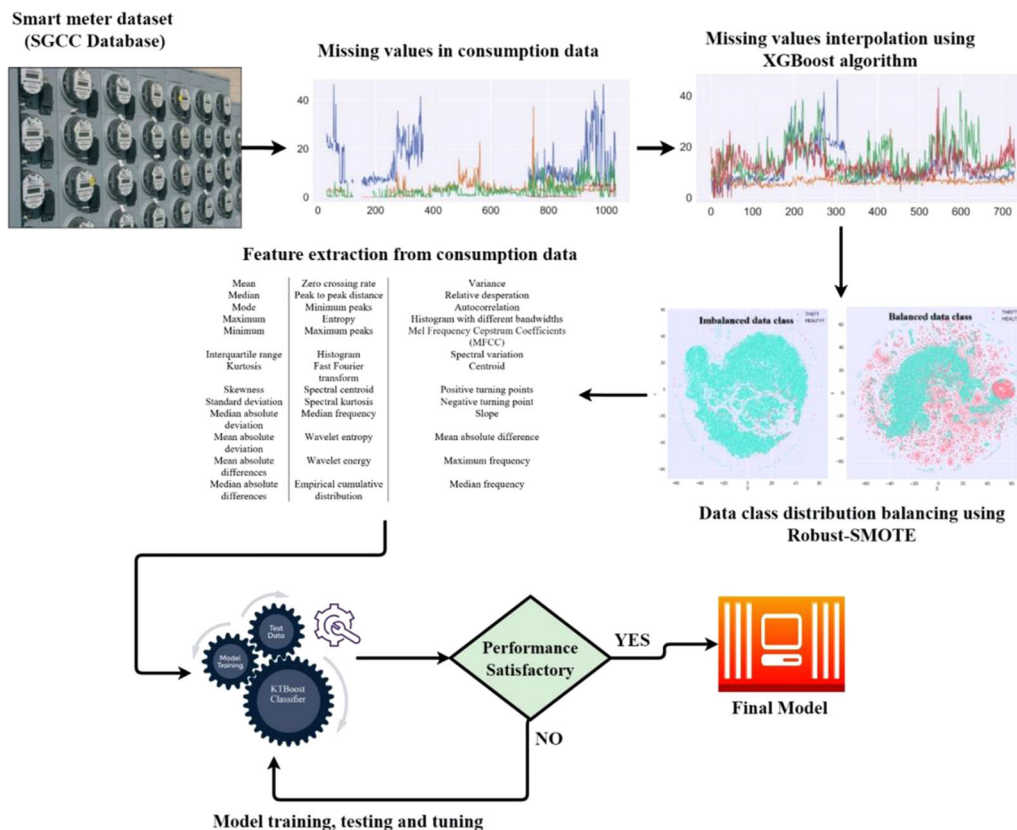


FIGURE 1 Proposed Jaya optimized-KTBoost based electric theft detection framework

The data from smart meters is often irregular, with several null and outlying readings mainly due to unstable synchronous transmission between sensors and databases, unexpected device maintenance, storage issues, unreliable/inadequate quality network, the incorrect estimate of sent data, and various unknown environmental factors [26]. Such irregularities in the dataset may jeopardize the learning ability of the SML classifier, resulting in biased and erroneous estimations [27]. In order to address this issue, typically, two approaches have been adopted in literature: imputation or elimination. In the imputation method, an estimated value for the missing attribute is substituted, while in elimination, the missing entries in the dataset are removed. The imputation process is often used for dealing with missing features since it is based on the concept that if an essential feature is missing for a specific instance, it may be approximated from the already available data [28]. In general, the imputation process is carried out either by statistical or machine learning methods. The estimation techniques are based on statistical methods such as mean, mode, median, linear interpolation [29], or autoregressive integrated moving average [30]. These data imputation methods are computationally fast and simple to execute. However, they generally lead to erroneous and skewed results due to the possible presence of outliers (individuals or observations with unusual characteristics) in the data. Furthermore, most of the classifiers cannot comprehend the complex relationships between input data variables and missing values

occurrence patterns in the data, which consequently leads to misleading outcomes. Nevertheless, few machine learning methods such as k-nearest neighbour missing values imputer [31], fuzzy clustering [32], support vector regressor (SVR) [33], random forest imputation (RFI) [34], Bayesian missing values imputer [35], etc., employ efficient predictive modelling techniques for estimating missing data values accurately. However, in the presence of huge amounts of data, such as the high-resolution data from smart meters, the mentioned techniques require enormous computing resources. Another way to deal with missing data is to discard/eliminate it entirely from the rest of the data. Despite the fact that “discarding” techniques such as list and pair-wise can be implemented smoothly, a significant loss of information might happen, leading to skewed estimates at the end of the classification process.

Another challenge in NTL detection is the unbalanced data class distribution, that is, the frequency of fraudulent cases is disproportionately low compared to genuine consumer cases. The performance of machine learning classifiers is severely affected by the imbalanced distribution of data classes. Moreover, the over-representation of the majority class (Healthy consumers) prevents a classifier from focusing on minority class (Fraudster customers); thus, producing irrational results. Various methods based on the concepts of minority oversampling and majority under-sampling have been proposed in the literature to counteract this issue. Two prominent research works

that have thoroughly addressed this imbalanced data class distribution problem are Nazmul et al. [36] and Sravan et al. [37]. Both works used the Synthetic Minority Oversampling Method (SMOTE) to balance the data class distribution in the acquired NTL detection dataset. The SMOTE method randomly generates the minority class samples by setting the same sampling rate for all samples of the minority class. The problem associated with this approach is that it causes overfitting and low generalizing ability of the classifier. In another research work, Madalina et al. [38], an under-sampling method is employed where the number of data samples from the majority class is eliminated to balance the data class distribution. Such data balancing methods are simple to execute; however, they can cause significant data loss, resulting in a reduction in the accuracy of the developed model. In another article [39], the data class distribution was balanced via the use of the ADaptive SYNthesis (ADASYN) based oversampling technique. While the developed approach obtained better generalizing ability, it achieved lower accuracy owing to the underfitting of the developed model.

As mentioned earlier in this section, the third major problem in the fraud detection techniques is the selection of the most relevant features for the model training process. Due to the fact that raw smart meters contain only consumption data and lack any statistical or supplementary features, it becomes difficult for the learning classifier to differentiate/understand the complex underlying patterns present in the data. In order to mitigate this issue, Punmiya et al. [40] and Salman et al. [24] extracted additional features from raw data employing simple statistical techniques such as mean, median, standard deviation, minimum and maximum. However, even though these techniques are simplistic to implement and computationally fast yet, they produce misleading results in the presence of outliers in the data.

After feature engineering, choosing a suitable classifier for efficiently separating genuine and fraudulent customers is the next challenge in any supervised ML technique. Nagi et al. [39] used a predictive modelling technique based on support vector machines (SVM) to identify abnormal behaviour of the consumers. The SVM-based ML model was developed using customer load profile data and other characteristics such as creditworthiness rating, meter reading data, and fraudulent activity report to identify abnormal consumer behaviour effectively. However, the detection hit rate achieved was merely 60% which is significantly very low, particularly when consumers are in the millions. In one of the most recent studies, a deep Siamese network (DSN) coupled with a convolutional neural network (CNN) and long-short term memory (LSTM) was proposed by Javaid et al. [39] to differentiate the characteristics of genuine and dishonest consumers. The authors achieved a reasonable accuracy; however, the precision and recall rates were comparatively lower. In another study, Paria et al. [41] developed a theft detection framework to identify regions of significant energy theft at the transformer level using data gathered from different distribution transformer meters. The developed methodology achieved a high detection rate (94%); however, since the fraudster consumption patterns introduced in this research work were produced synthetically, they do not pre-

cisely depict the actual fraudster customer's profiles; therefore, attained outcomes may diverge from a realistic scenario.

In one of the recent studies, Oprea et al. [42], utilized feature engineered light gradient boosting to effectively find irregular consumption patterns in the acquired conventional meter dataset. However, the data class balancing technique employed in the quoted study used the SMOTE algorithm, which is prone to overfitting and often results in a high generalizing error. In addition to that, it may increase noise since it ignores class distributions and has some sample selection blindness. Sarkar et al. [25] presented the fraud detection framework utilizing ensemble machine learning methods with considerable high accuracy, precision, and recall. However, they failed to interpret the developed model outcomes, which are crucial in strengthening the ML model further. The model's outcomes interpretation benefits in two ways: first, it helps concentrate and fine-tune the characteristics that contributed most to generating positive outcomes. Second, by re-training the model with a smaller set of very important features (features importance score assigned by the model), computational time may be substantially lowered without compromising real accuracy values. Table 1 presents the summary of the different techniques utilized in developing SML-based electric theft detection methods.

### 3 | PROPOSED METHODOLOGY

A stage-wise representation of the proposed theft detection framework is depicted in Figure 2.

Each of the stages mentioned in Figure 2 is detailed in subsequent subsections.

#### 3.1 | Exploratory data analysis

In this subsection, the pre-processing of the acquired dataset is explained in detail. The dataset used for this study is real smart meter data obtained from the State Grid Corporation of China (SGCC). The acquired dataset distribution is summarized in Table 2. Like most of the real-time datasets, the number of fraudster consumers in SGCC kWh data is lower than that of healthy consumers. Figures 3 and 4 illustrate the consumption patterns of a few randomly selected fraudulent and healthy consumers, respectively.

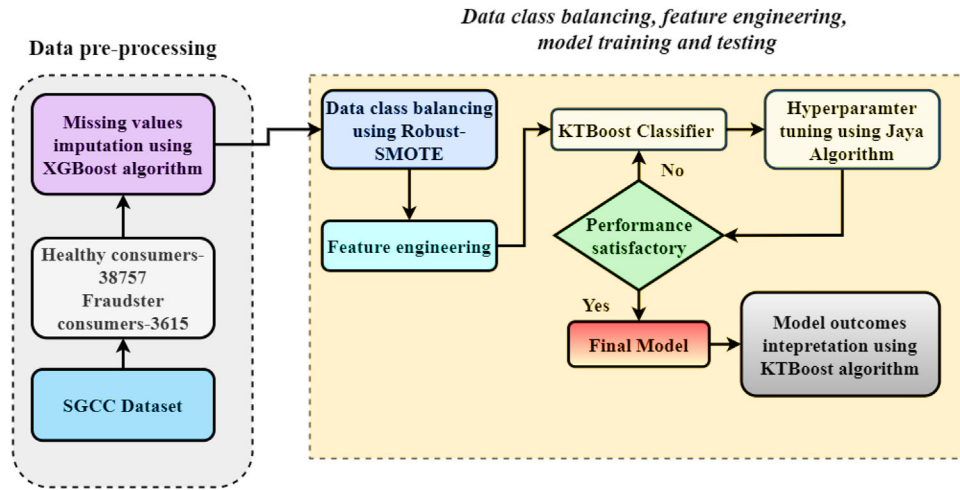
It can be observed from the provided figures that the consumption patterns of the theft customers are highly unpredictable and contain low repeatability, while the genuine consumers' patterns are recurrent and exhibit a relationship among identical periods of subsequent years.

#### 3.2 | Missing values and their imputation using XGBoost algorithm

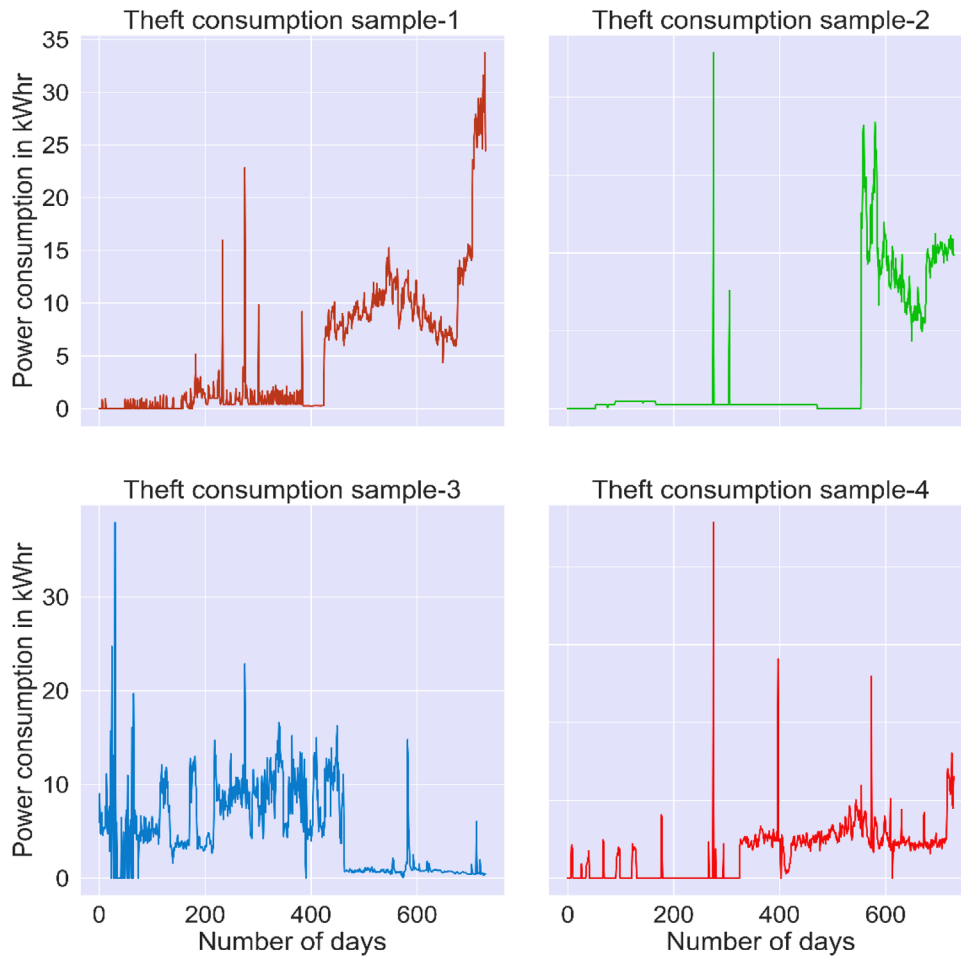
The smart meter data often contains numerous missing entries mainly due to the malfunction of equipment, lag in

**TABLE 1** Summary of most widely used techniques in building SML based electric theft detection methods

S. No.	References	Method used	Missing values	Data class imbalance	Feature extraction	Feature selection	Performance metrics utilized
1	Nizar et al. [43]	Naïve Bayes and Decision tree	–	–	Load profiles	–	Accuracy
2	Nagi et al. [44]	Genetic algorithm-SVM	Average values	–	Statistical features	–	Accuracy, detection rate
3	Nizar et al. [45]	Extreme learning machine -SVM	–	–	–	–	Accuracy
4	Nagi et al. [46]	SVM	Average values	–	Statistical features	–	Accuracy, detection rate
5	Ramos et al. [47]	Optimum path forest (OPF)	–	–	Statistical features	–	Accuracy
6	Caio et al. [48]	Harmony search algorithm and OPF	–	–	Principal component analysis	Harmony search algorithm	Accuracy
7	Carlos et al. [49]	Integrated expert system, rule-based system	Removal	–	Text mining	–	Accuracy
8	Faria et al. [50]	Spatial-temporal estimation	–	–	Statistical features	–	Loss probability
9	Juan et al. [51]	SVM-DT	–	–	Statistical features	Filter wrapper	Accuracy, recall, precision, and $F1_{score}$
10	Paria et al. [52]	Consumption pattern-based energy theft detection	–	Different sampling proportions	Statistical features	–	Bayesian detection rate, accuracy, recall, detection rate, and precision
11	Selvam et al. [53]	Decision Tree, Random Forest	–	–	–	–	Accuracy, ROC
12	Zheng et al. [54]	Wide and deep convolutional neural networks	Average values	–	CNN	–	Accuracy, recall, detection rate, and precision
13	Punmiya et al. [40]	Feature engineered extreme gradient boosting machine	–	SMOTE	Statistical features	–	Accuracy, recall, detection rate, and precision
14	Salman et al. [13]	Ensemble machine learning	–	–	–	–	Accuracy, recall, detection rate, and precision
15	Blazakis et al. [55]	Adaptive Neuro-Fuzzy Inference System	–	–	Statistical features	Neighbourhood component analysis	Accuracy, F1 score, precision, recall, specificity, AUC
16	Sravan et al. [25]	Ensemble machine learning	Deletion	SMOTE	–	–	Accuracy, ROC, recall, precision
17	Salman et al. [24]	Boosted C5.0 decision tree	–	–	Statistical features	Pearson's Chi-Square	Accuracy, recall, detection rate, and precision
18	Zhengwei et al. [56]	Random Forest	–	Kmeans-SMOTE	–	–	Accuracy, TPR, FPR, TNR, G-mean
19	Guoying et al. [57]	Autoencoder and Random Forest	–	Undersampling and re-sampling	Stacked autoencoder	–	Probabilistic prediction
20	Munwar et al. [58]	Recurrent neural network	Rule-based	–	–	–	Accuracy, recall, detection rate, and precision
21	Cheng et al. [59]	Deep learning, random forest	Rule-based	–	CNN	–	Precision, recall, true positive rate, false-positive rate
22	This work	Jaya optimized-KTBoost	XGboost algorithm	Robust-SMOTE	Statistical, temporal, and spectral domain-based features	KTBoost algorithm	Accuracy, detection rate, precision, F1score, kappa and MCC



**FIGURE 2** Proposed Jaya optimized KTBoost based electric theft detection framework



**FIGURE 3** Electric consumption patterns of fraudster consumers

registering/collection of data remotely, accidental deletion, cyber-attacks or fabrication of their smart meter devices, etc. In order to illustrate the occurrence of the missing values in consumption patterns, a few consumer's electric power con-

sumption randomly sampled from acquired consumption data are illustrated in Figure 5.

From Figure 5, it can be observed that there are several blank spots in between the consumption values. If such kind

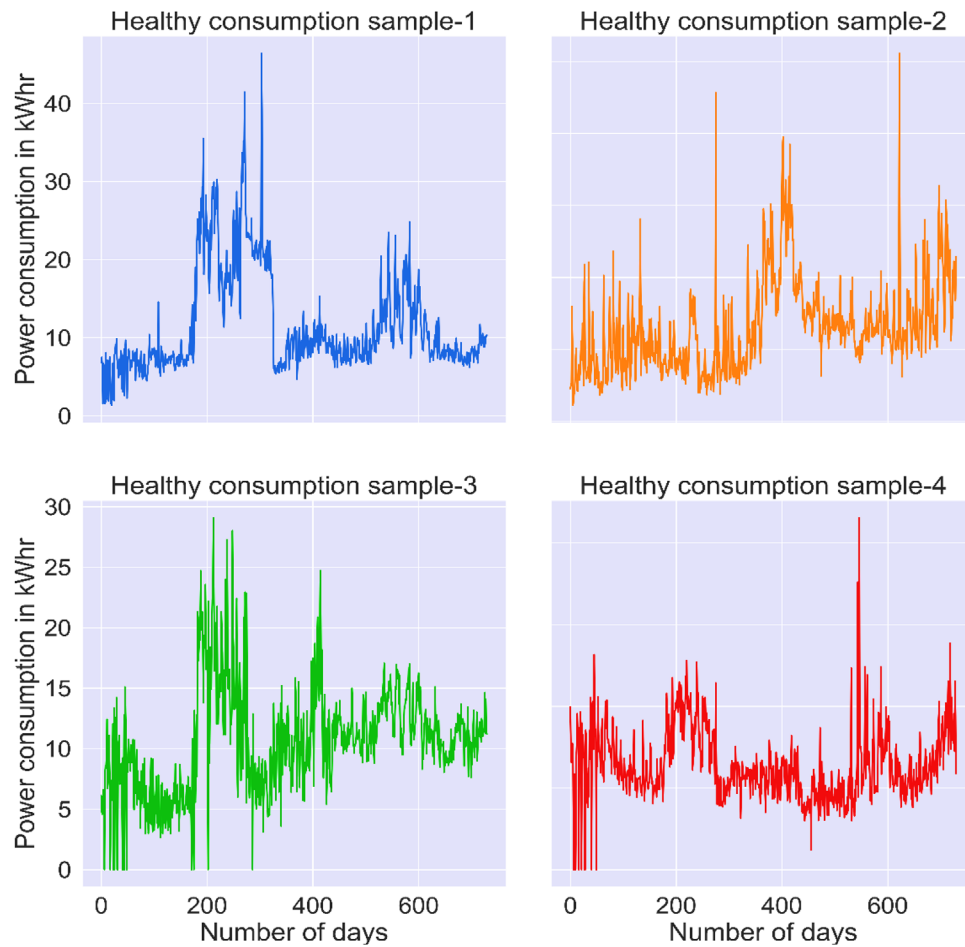


FIGURE 4 Electric consumption patterns of healthy consumers

TABLE 2 Data statistics of acquired SGCC dataset

Parameter description	Parameter value
Number of total consumers	42,372
Number of healthy/genuine consumers	38,757 or 91.46% of total data
Number of fraudster/theft consumers	3615 or 8.54% of total data
Number of days of consumption record	1035 days (January 2014 to December 2016)

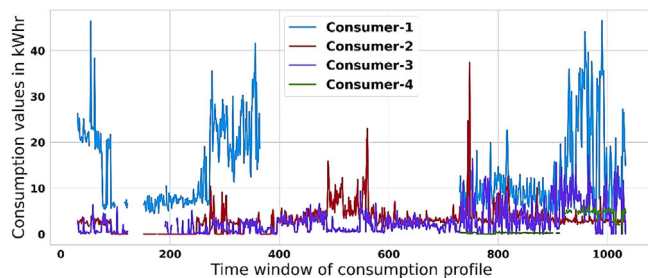


FIGURE 5 Randomly samples consumers' consumption data with missing entries

of incomplete dataset is directly fed into the ML framework, the ML algorithms within the framework would be unable to comprehend the complicated relationships between input data variables and missing values occurrence patterns present, thus leading to misleading conclusions. The missing values in the entire dataset are computed and plotted in Figure 6. Figure 6 illustrates the missing values present in each consumer's consumption data where the x-axis is the time window of acquired consumption data, and the y-axis is the number of consumers present in the data. The darker regions in the mentioned figure demonstrate a higher density of missing entries, and lighter or dotted areas express lesser missing entries. For example, from the time window of 2014 to 2015, consumers' consumption data carries a lot of missing entries, whereas, in 2016, these missing entries are comparatively lower. In addition to that, the kernel density estimation and histogram plot of missing values present in the data is computed and illustrated in Figure 7.

It may be noted from Figure 7 that there are more than 7000 consumers whose missing value count is greater than 700, while the same count for the majority of the consumers is in between 10 to 200. To address this issue, the proposed framework utilizes a machine learning-based technique to build a predictive model employing the XGBoost algorithm for estimating the missing

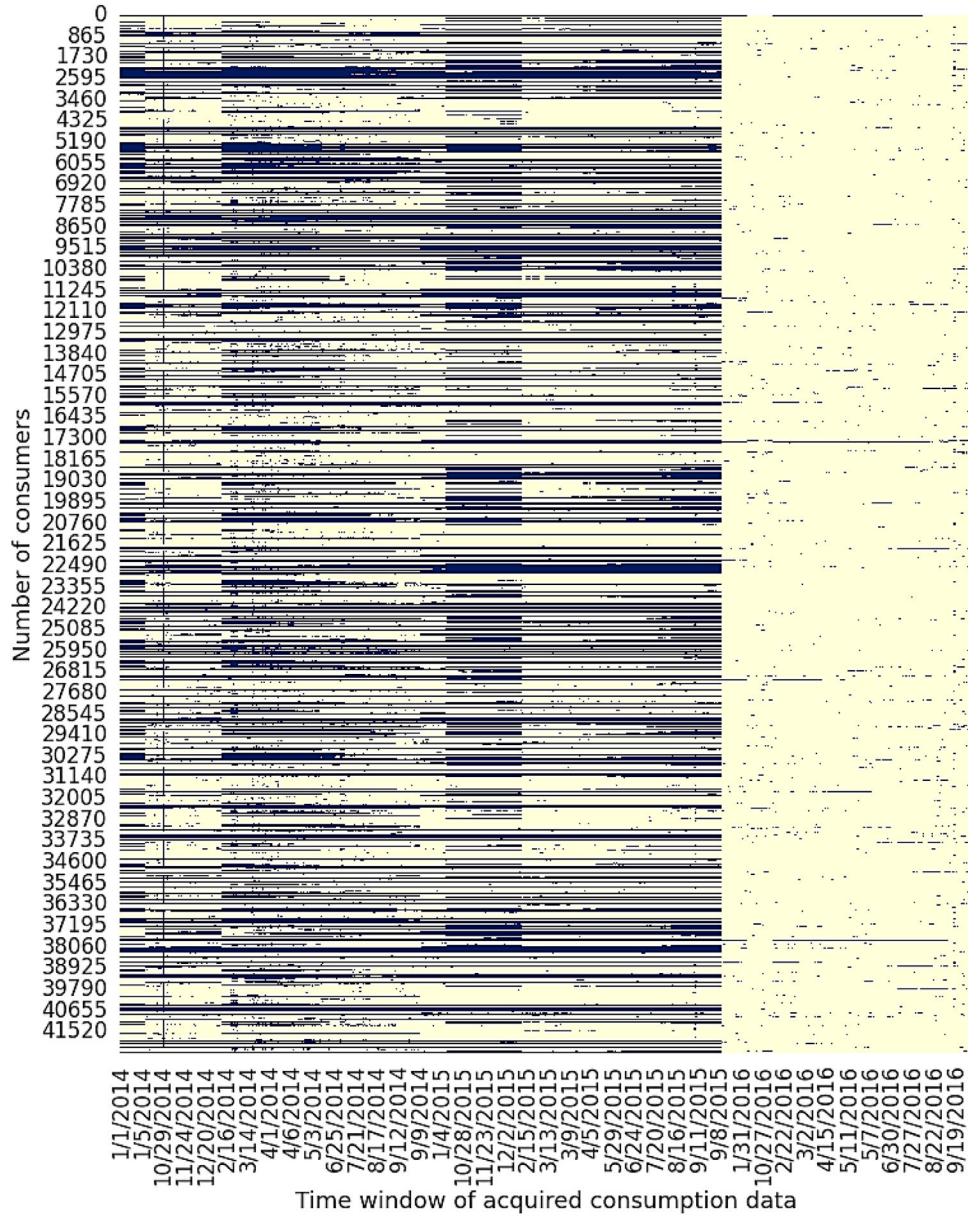


FIGURE 6 Missing values occurrence in the acquired smart meter (SGCC) dataset

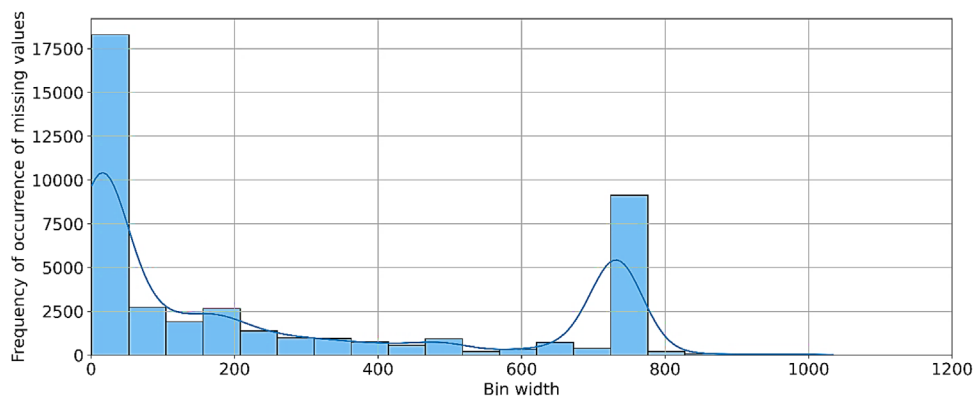


FIGURE 7 Histogram-Kernel density estimation plot of missing values present in acquired smart meter dataset



attributes present in the data. The XGBoost algorithm is one of a group of ensemble machine learning algorithms that use the decision tree-based boosting technique to generate the most accurate models/estimators. In addition, it can impute missing entries present in a dataset, adaptable to interactions and non-linearity within data, and scalable to large data situations. The boosting technique in the XGBoost refers to the process of progressively creating multiple models where each newly created model attempts to fix the error in the preceding model. XGBoost utilizes the decision tree as a base classifier and progressively builds each subsequent new decision tree based on the prediction results of the previous decision trees. The overall objective function of the XGBoost algorithm is given in Equation (1).

$$\begin{aligned} & \text{Objective function}(\theta) \\ & = \sum_j \text{TrainingLoss}(\hat{y}_j, y_j) + \sum_i \lambda(f_i), f_i \in F \quad (1) \end{aligned}$$

where  $y_j$  is the actual value and  $\hat{y}_j$  is a prediction made by the model. The training loss here controls the overall performance of the models. The regularization function  $\lambda$  computes the complexity of the model, which further assists in preventing the model from overfitting.  $F$  represents the function space where the set of all possible regression tree functions ( $f$ ) occurs. The current research work utilizes the intelligence of the XGBoost algorithm for imputing missing entries in the acquired dataset. To visualize the data imputation process, the missing values for two of the randomly selected samples from the acquired dataset are imputed using the mentioned algorithm. The results attained by the proposed missing values imputation technique are provided in Figure 8.

It can be observed from Figure 8, the estimated missing values (in black colour) coincide with the actual consumption data. Thus, the missing values imputed through this process enhance the ML classifier performance and avoid unintentional model bias towards the missing values.

### 3.3 | Robust-SMOTE for data class imbalance issue

The SML-based classifier's performance deviates largely if the proportion of data classes present in the acquired dataset varies [60]. Since the acquired smart meter data is highly unbalanced, class balancing must be performed through an intelligent technique before training and testing the classifier. Figure 9 shows the class distribution of the collected dataset; the red data points represent the theft samples and green points healthy samples (majority class).

It can be observed in Figure 9 that the minority class samples are scarcer than the majority class samples. The ML-classifiers trained on such datasets are likely to be biased towards the data class that is present in a greater proportion. Generally, legitimate customers are more than fraudsters in most of the smart meters

dataset [42]. Therefore, it is essential to balance the distribution of the data classes prior to feeding the ML-classifier.

In order to mitigate this issue, the robust SMOTE algorithm is used in this study. The robust SMOTE method addresses all frequently occurring categories of minority data samples, that is, minority points in the majority class region, minority class close to majority class samples, and safe minority points [61]. It accomplishes the mentioned task by measuring the relative data density for computing the local density of the minority data points between its  $k$ -nearest heterogeneous neighbours and  $k$ -nearest homogeneous neighbours initially. Afterward, it divides minority samples into borderline and safe samples relying on the relative density of minority samples' 2-means clustering outcomes. The quantity produced by each minority data point is re-weighted depending on the number of majority classes in its  $k$ -nearest neighbours, resulting in more samples close to the safe data points. In comparison, the scarcer samples are brought near the disorder samples to improve the divisibility of the classification boundary between classes. The data class distribution of the acquired dataset after implementing the robust-SMOTE is illustrated in Figure 10.

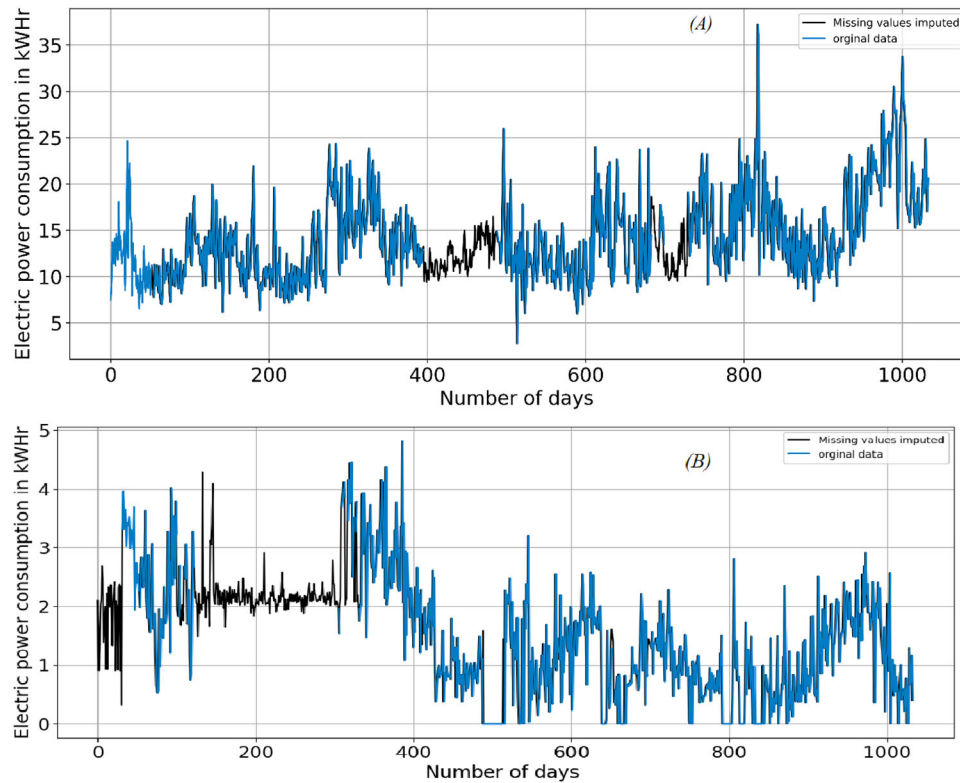
It can be observed from Figure 10 that the minority (red data points) and majority class (green data points) distribution is justifiably balanced. Furthermore, most of the minority class samples are generated from those safe minority samples that are far away from the healthy samples; thus, this method aids the ML-classifier in defining the classification border more eloquently.

### 3.4 | Feature engineering

The successful development of the ML model is often contingent on the appropriate selection of input features used during model training [62]. The feature engineering approach is specifically dedicated to that purpose; it assists in summarizing the dynamics of the data and enhances its overall representation by extracting the most important features while simultaneously improving the performance and detection accuracy of the model [63]. The acquired smart meter dataset consists of only consumption data in kWh and lacks any other statistical significance. Therefore, in this study, several features from statistical, temporal, and spectral domain-based features are extracted from each consumer's consumption data, as presented in Table 3. Since there are no less than 39 extracted features presented in Table 3, therefore, it is quite hard to add the theoretical and mathematical background of all the extracted features due to the scope and length of the article. Nevertheless, interested readers can find all the relevant information in reference [64].

### 3.5 | Proposed classifier: Jaya optimized KTBoost algorithm

Boosting algorithms are widely used in practical data science and machine learning-based research works due to their outstanding



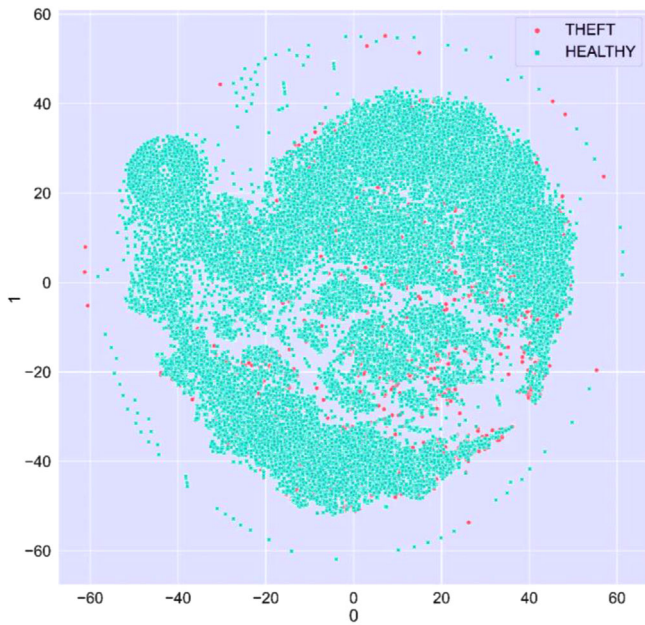
**FIGURE 8** (A, B) Missing values imputation in consumer's consumption data using XGBoost algorithm

**TABLE 3** Extracted features from time-series data

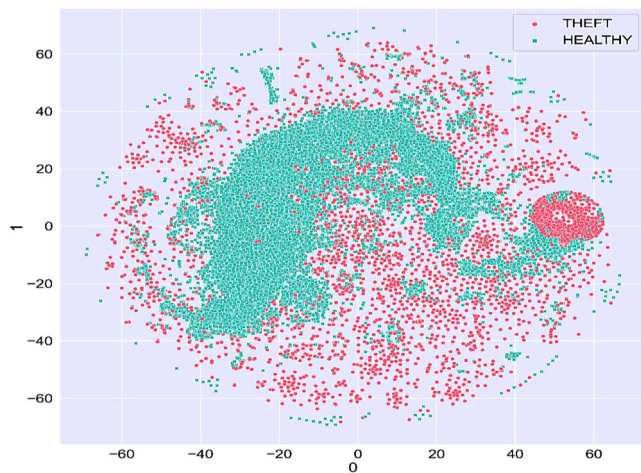
S. No.	Feature	S. No.	Feature	S. No.	Feature
1	Mean	14	Zero crossing rate	27	Variance
2	Median	15	Peak to peak distance	28	Relative desperation
3	Mode	16	Minimum peaks	29	Autocorrelation
4	Maximum	17	Entropy	30	Histogram with different bandwidths
5	Minimum	18	Maximum peaks	31	Mel frequency cepstrum coefficients (MFCC)
6	Interquartile range	19	Histogram	32	Spectral variation
7	Kurtosis	20	Fast Fourier transform	33	Centroid
8	Skewness	21	Spectral centroid	34	Positive turning points
9	Standard deviation	22	Spectral kurtosis	35	Negative turning point
10	Median absolute deviation	23	Median frequency	36	Slope
11	Mean absolute deviation	24	Wavelet entropy	37	Mean absolute difference
12	Mean absolute differences	25	Wavelet energy	38	Maximum frequency
13	Median absolute differences	26	Empirical cumulative distribution	39	Median frequency

prediction accuracy on highly complex datasets [65]. The boosting algorithms additively chain weak (base) classifiers by consecutively reducing both bias and variance at each boosting iterations. Despite the widespread usage of boosting algorithms, only one type of function is used as a base learner in most cases. In contrast to that, the KT-Boost algorithm either adds a regression tree or a penalized reproducing kernel

Hilbert space RKHS (kernel ridge regression function) to the ensemble of base classifiers in each boosting iteration [66]. In the beginning, the base learner is learned from both regression tree and RKHS function by employing gradient or newton as optimization techniques; afterward, the base learner whose inclusion in the ensemble results in the lower empirical risk is chosen. In this way, at each subsequent iteration, a base learner



**FIGURE 9** The unbalanced data class distribution in obtained smart meter dataset



**FIGURE 10** The balanced data class distribution after robust-SMOTE algorithm

from two fundamentally different learners is selected to achieve high predictive accuracy. In addition to that, this amalgamation facilitates enhanced learning about functions that have different regularity degrees, such as discontinuities and smooth portions, as most discontinuities portions are learned through regression trees through smooth (continuous) portions using RKHS regression functions. The most important hyper-parameters of the KTBoost algorithms are given in Table 4.

Unlike the previous research work where these parameters are either selected by using inefficient and time-consuming “trial and error” method or are adopted from previous literature, the current study utilizes the intelligence of a swarm intelligence based optimization technique called the Jaya algorithm to select the most optimal hyperparameters of the KTBoost

**TABLE 4** Hyperparameters of the KTBoost classifier

Parameter name	Description
learning_rate	Parameter helps in setting weighting factors for the addition of new trees at each iteration to the classifier.
n_estimations	The number of boosting iterations to be performed.
subsample	The number of samples to be used for fitting the individual base learners. Optimal selection of this parameter can assist in setting bias and variance values.
criterion	This is an evaluation metric to compute the quality of split, by default, it is selected as the mean square error (mse) but can be chosen as mean absolute error or Friedman mse.
min_samples_split	The minimum number of samples to be present at a leaf/internal node. This parameter controls the model overfitting/ underfitting related problems.
min_samples_leaf	The minimum number of samples to be present at the leaf. Controlling this parameter helps in overfitting/underfitting related issues.
min_weight_leaf	
max_depth	Parameter helps in building the structure of regression tree.
max_features	Number of features to be selected when searching for split.
max_leaf_nodes	Optimal selection of these value facilities reducing the impurity of regression trees.
base_learner	This parameter sets the base learners, in this either trees or kernel or a combination of both can be chosen.
update_step	This parameter estimates boosting updates at each iteration. If the base learner is chosen only trees and update step as a hybrid then gradient step estimates the structure of trees and Newton step assists in finding the number of the leaf. Similarly, if the base learner is chosen kernel and update step as a hybrid, then gradient descent is used as an update step.
Tol	This value facilities for early stopping if there is no change in the loss.
kernel	In the case of kernel booting, Laplace, radial basis function and generalized Wendland can be chosen as kernel functions.
range_adjust	Regularization parameter for RKHS regression function.
Nystroem	The Nystroem sampling method is used if set to true. In the case of large data set, this parameter helps in reducing computation resources.
n_components	The number of samples used in Nystroem samples.

algorithm. The Jaya algorithm is a gradient-free metaheuristic optimization method for solving constrained and unconstrained optimization problems. It is a stochastic population-based technique that modifies a population of individual solutions on an ordered basis by keeping the notion that each individual solution strives to attain the best solution while avoiding the least fit (worst) solution. One of the important features of this algorithm that makes it different from the

other swarm intelligence-based optimization methods is that it does not require any algorithm-specific or control parameters for its operation. To avoid the computational complexity and to achieve the most optimal results within the limited number of iterations, only eight of the most important hyperparameters (base\_learner, kernel, learning\_rate, loss, max\_depth, max\_leaf\_nodes, n\_neighbors, update\_step) are taken as decision variables in the current research work.

To achieve the best solution, the Jaya algorithm undergoes the following sequential steps,

Step 1: Initialize the input parameters of Jaya ( $Pob_{size}$ ,  $It r_n$ ) and of the problem which is to optimize ( $V ar_n$ ). In this the  $Pob_{size}$  is the population size.  $It r_n$  is the number of maximum iterations to set  $V ar_n$  is the design variables of the function which is to be optimized.

Step 2: Initiate by randomly initializing the population within the predetermined lower and upper boundaries as given as in Equation (2),

$$S_{ij} = S_{min,j} + (S_{max,j} - S_{min,j}) \cdot rand(0, 1) \quad (2)$$

where,  $S_{ij}$  is solution vector ( $S_{i1}, S_{i2}, S_{i3}, S_{i4}, \dots, S_{in}$ ),  $j = 1, 2, 3, 4, \dots, n$  (number of given design variables) and  $i = 1, 2, 3, \dots, Pob_{size}$  (total number of search agents).  $S_{max,j}$  Upper bound and  $S_{min,j}$  lower bounds of design variables.

Step 3: For each solution vector, estimate the value of the cost function and compute the best and worst solutions.

Step 4: Update the solutions as follows

$$S_{i,j,m}^{updated} = S_{i,j,m} + \alpha_{1,j,m} (S_{i,best,m} - |S_{i,j,m}|) - \alpha_{2,j,m} (S_{i,worst,m} - |S_{i,j,m}|) \quad (3)$$

where  $\alpha_1, \alpha_2$  are the two random numbers in between (0, 1) assisting in achieving the right balance between the exploration and exploitation process. The term  $\alpha_{1,j,m} (S_{i,best,m} - |S_{i,j,m}|)$  leads towards the worst solution whereas  $\alpha_{2,j,m} (S_{i,worst,m} - |S_{i,j,m}|)$  leads towards the best solution.

Step 5: Evaluate the updated solutions by restricting them not to exceed the boundary conditions.

$$S_{i,j,m}^{updated} = \begin{cases} S_{max,j} & \text{if } S_{i,j,m}^{updated} > S_{max,j} \\ S_{min,j} & \text{if } S_{i,j,m}^{updated} < S_{min,j} \\ S_{i,j,m}^{updated} & \text{otherwise} \end{cases} \quad (4)$$

Step 6: To evaluate whether the updated solution or the existing solution will advance to the next iteration, compute the value of the costs function for each set of search agents by employing the greedy selection technique. If the revised solution is better than the cur-

rent solution, replace the former. On the contrary, the revised solution will be discarded, but the current solution will be retained in the population.

## 4 | RESULT AND DISCUSSION

In this section, the performance of the proposed theft detection framework is evaluated and compared against the latest ML techniques such as XGBoost, lightGBM, Extra Trees classifier, and traditional ML techniques such as SVM, logistic regression, KNN, Ridge classifier, Linear discriminant classifier, and Naive Bayes classifier. In supervised ML learning, the trained classifiers are validated based on their ability to effectively predict and generalize the unlabelled data. In order to accomplish this task, various performance metrics exist, as mentioned in this study [10]. However, it is not practical to assess and analyse all of the metrics specified in the study; thus, few of the most relevant metrics are considered, as noted below.

$$\text{Accuracy} = \frac{T^+ + T^-}{T^+ + T^- + F^+ + F^-} \quad (5)$$

$$\text{Recall or detection rate} = \frac{T^+}{T^+ + F^-} \quad (6)$$

$$\text{False - positive rate} = \frac{F^+}{F^+ + T^-} \quad (7)$$

$$\text{False - negative rate} = \frac{F^-}{F^- + T^+} \quad (8)$$

$$\text{Precision or positive predictive value} = \frac{T^+}{F^+ + T^+} \quad (9)$$

$$F_1 - \text{score} = \frac{2T^+}{2T^+ + F^+ + F^-} = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

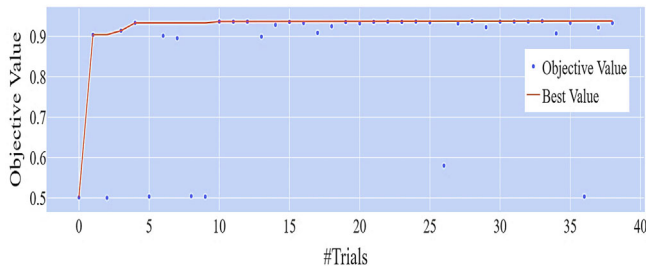
Matthews correlation coefficient (MCC)

$$= \frac{T^+ * T^- - F^+ * F^-}{\sqrt{(T^+ + F^+) (T^+ + F^-) (T^- + F^+) (T^- + F^-)}} \quad (11)$$

$$\text{Kappa value} = \frac{\rho_0 - \rho_e}{1 - \rho_e} \quad (12)$$

where  $T^+$  is the true positive,  $T^-$  is the true negative,  $F^+$  is the false positive and  $F^-$  is the false negative.  $\rho_0$  predicted value and  $\rho_e$  actual value.

At this stage, the dataset developed during the feature engineering process is retrieved for model training and validation purposes. The fetched dataset comprises 1035 days of real consumption data and 39 additional features (mentioned in Table 3). Moreover, the raw input dataset's data class distribution was balanced with the robust-SMOTE method prior to feeding it to the algorithm for model training. The train-test



**FIGURE 11** The proposed model’s accuracy values against several optimization trails

**TABLE 5** Optimal hyperparameters

Hyperparameter	Value
<i>base_learner</i>	Combined (Kernel boosting and tree boosting)
<i>kernel</i>	GW
<i>learning_rate</i>	0.2
<i>loss</i>	deviance
<i>max_leaf_nodes</i>	34
<i>max_depth</i>	1863
<i>n_neighbors</i>	50
<i>update_step</i>	hybrid

split method is used in which 80% of the data is used for model training while 20% is for testing purposes. The proposed theft detection framework utilizes the KTBoost algorithm for model training, while the Jaya algorithm-based meta-heuristic optimization is used for its hyper-parameter tuning. In this scenario, the objective function for optimization purposes is to optimize the model’s accuracy by minimizing the difference between predicted and actual outcomes. By initializing more than 35 trails/iterations employing the Jaya algorithm model attained an accuracy of 0.937 as presented in the optimization history plot in Figure 11. The *x*-axis represents the trail count, while the *y*-axis shows the accuracy value. The blue dots show the accuracy value attained at different combinations of hyperparameters in the graph.

Furthermore, in Figures 12 and 13, the slice and contour plots of the model’s hyperparameters optimization process are shown, neatly illustrating the implication of the hyper parameter’s variation on the objective value/accuracy. For example, Figures 12 and 13 depict that a learning rate within the range of 1.5 to 2.5 achieves high objective values, but increasing beyond that produces a considerable reduction in objective value. Similarly, *max\_depth* greater than 1500 yields better accuracy values; increasing beyond that yields a significant reduction in accuracy, which can be attributed to the model overfitting on the training data.

The optimal hyper-parameters set, which attained the best accuracy value during several optimizations trials, is given in Table 5. As presented in the table, the combined base learner (kernel boosting and tree boosting) and hybrid update step achieve the best accuracy value.

### 4.1 | K-fold cross-validation results of the Jaya optimized-KTBoost model

To effectively implement the proposed Jaya optimized-KTBoost algorithm, the designed model is initially trained on the data developed after the data class balancing and feature engineering stage. Afterward, the tenfold cross-validation (CV) technique employing the mentioned performance metrics (Equations (5)–(12)) is utilized for the performance evaluation of the designed model. This evaluation has produced the following results; as presented in Table 6, the proposed model has achieved a mean accuracy and precision of 0.9338 and 0.9508 with a standard deviation (SD) of 0.0029 and 0.0035, respectively.

### 4.2 | Confusion matrix evaluation of the proposed model

The confusion matrix (CM) is a prominent metric for addressing classification issues. It may be used for both binary classification and multiclass classification issues. CM represents counts from the actual and predicted values, as illustrated in Figure 14. In this study,  $T^+$  represents the number of theft consumers rightly classified by the classifier whereas  $F^-$  represents the fraudster consumers misclassified as the healthy consumers. Similarly,  $T^-$  represents the number of rightly classified healthy consumers while  $F^+$  depicts the healthy consumer misclassified as the fraudster consumer.

The confusion matrix of the proposed model is shown in Figure 15, “0” represents here the actual negative class or Healthy consumers and “1” represents the positive class or Fraudster consumer. The values in CM are normalized in the percentage form for ease in readability purposes. From the mentioned figure, it can be observed that the classifier rightly classified 93.16% of the theft consumers while 6.84% of actual theft consumers were misclassified as healthy. Similarly, 95.25% of healthy consumers were rightly classified, whereas 4.75% of actual healthy consumers were misclassified as theft.

### 4.3 | AUC-ROC curve of the proposed model

The receiver operating characteristic (ROC) is an important performance metric for evaluating binary classification algorithms [67]. It represents the trade between the true positive rate and the false-positive rate of the classifier in a bi-dimensional plot. The area under the ROC curve can be computed using Equation (13),

$$\begin{aligned}
 & \text{Area Under the Curve (AUC)} \\
 &= \frac{\sum_{j \in \text{positiveTarget}} \text{Rank}_j - \frac{P_s(1 + P_s)}{2}}{P_s * N_s} \quad (13)
 \end{aligned}$$

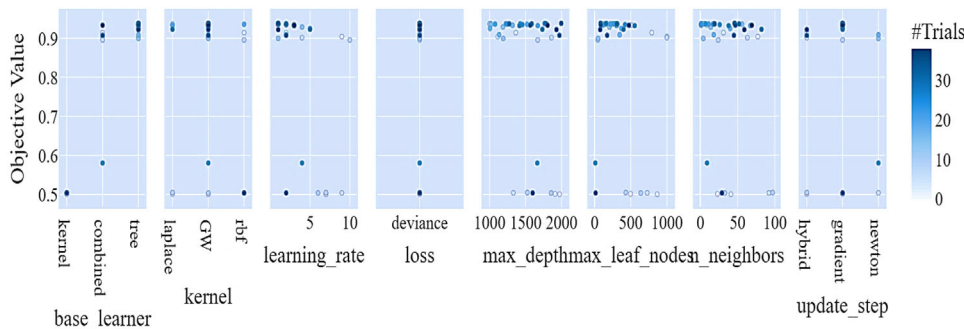


FIGURE 12 Slice plot of the proposed model against several optimization trails

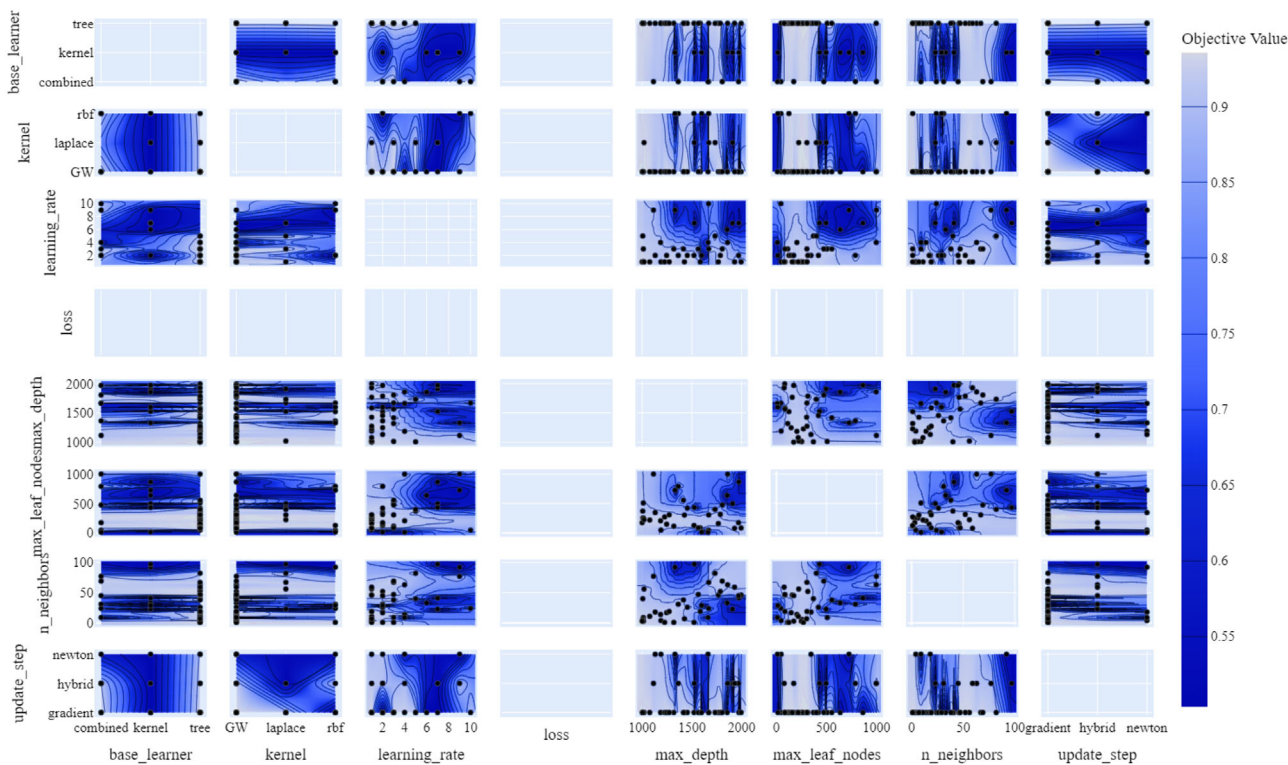


FIGURE 13 The contour plot of the proposed model against several optimization trails

TABLE 6 Jaya optimized-KTBoost model tenfold-cross validation results

No. of folds	Accuracy	Recall	Precision	F1 <sub>score</sub>	Kappa-value	MCC
1	0.9311	0.9216	0.9479	0.9345	0.8891	0.8922
2	0.9354	0.9278	0.95	0.9388	0.8705	0.9108
3	0.9354	0.9239	0.9536	0.9385	0.8706	0.9111
4	0.9326	0.9196	0.9524	0.9357	0.8921	0.9123
5	0.937	0.9263	0.9542	0.94	0.8736	0.9201
6	0.939	0.9292	0.9552	0.942	0.8777	0.9021
7	0.9285	0.9191	0.9454	0.9321	0.8921	0.9154
8	0.9331	0.9258	0.9476	0.9366	0.881	0.9125
9	0.9313	0.923	0.947	0.9348	0.887	0.8926
10	0.9344	0.9206	0.9548	0.9374	0.8891	0.9092
<b>Mean</b>	<b>0.9338</b>	<b>0.9318</b>	<b>0.9508</b>	<b>0.9371</b>	<b>0.8873</b>	<b>0.9077</b>
Standard deviation	0.0029	0.0033	0.00365	0.00292	0.0087	0.00931

	Predicted Negative	Predicted Positive
Actual Negative	$T^-$	$F^+$
Actual Positive	$F^-$	$T^+$

FIGURE 14 Confusion matrix for binary classification problem

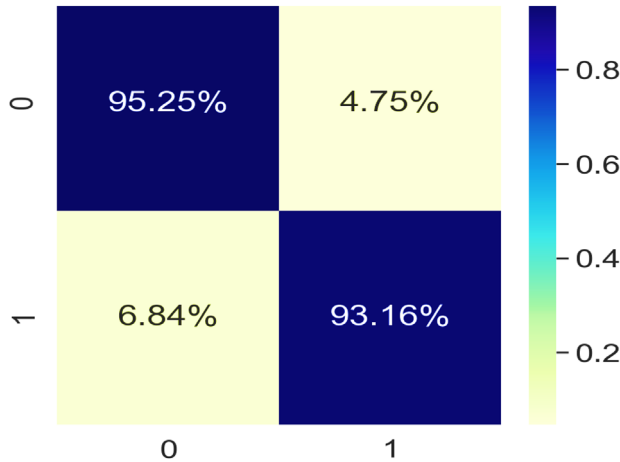


FIGURE 15 Confusion matrix of proposed theft detection model

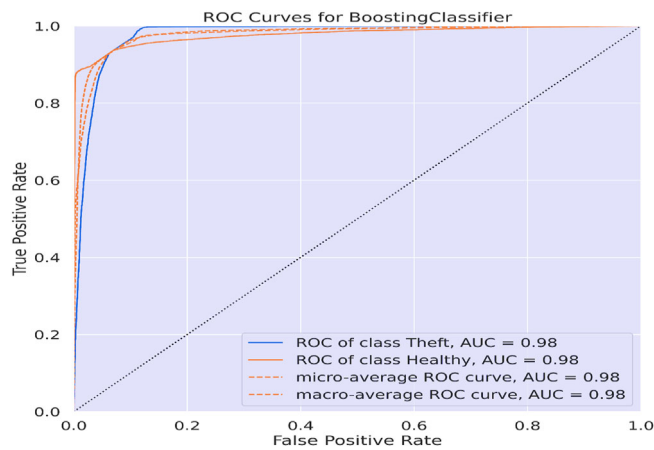


FIGURE 16 The ROC curve of the KTBoost classifier

where the  $P_j$  represents the number of positive samples,  $N_j$  number of negative samples and  $Rank_j$  depicts the rank value or of sample  $j$  belonging to the positive class. The AUC value is the likelihood that a randomly selected positive data sample would rank higher than a randomly selected negative data sample. The AUC value varies between 0.5 to 1, where 0.5 specifies that the classifier performs random guessing, and 1 indicates that the classifier is perfect in classifying the healthy and theft consumers.

The ROC curve of the proposed classifier is shown in Figure 16; the  $x$ -axis represents the FPR, and the  $y$ -axis the TPR. The average AUC value of the proposed classifier is 0.98, which indicates that most of the theft and healthy consumers are rightly classified.



FIGURE 17 The learning curve of the proposed theft detection model

#### 4.4 | The learning curve of the proposed theft detection model

A learning curve depicts the relationship between the training score and cross-validated (CV) test score for a classifier with different training data instances graphically [68]. The basic notion of this curve is to check the classifier’s generalizing ability on different data samples. The learning curve of the proposed classifier is shown in Figure 17. The curves in the graph illustrate the mean scores, while the shaded areas depict the standard deviations above and below the mean for all cross-validations. If the model is flawed because of the bias, the training score curve will most likely be more variable than expected. Likewise, if the model is prone to error owing to variance, the cross-validated score will be more unpredictable.

In Figure 17, it can be seen that when the data samples are minimal, the model training score is very high in comparison to the CV-score, which is a result of the high bias of the model. In contrast, as the number of training data samples grows, the training score decreases, while the CV- score increases, albeit with considerable fluctuation due to the model’s high variance. Additionally, it is interesting to note from the learning curve that the model’s CV-score and accuracy are above 0.9338, implying that the model can accurately distinguish fraudster consumers from healthy consumers.

#### 4.5 | Proposed model’s outcomes interpretation and their impact on training time

In this section, the proposed model’s prediction or outcomes are interpreted. The model’s prediction interpretation is the process by which the input data features utilized for model training are evaluated based on their positive influence on predicting the correct result. In this study, the KTBoost algorithm is employed to rank all the given input features in terms of their contribution in predicting the right outcome.

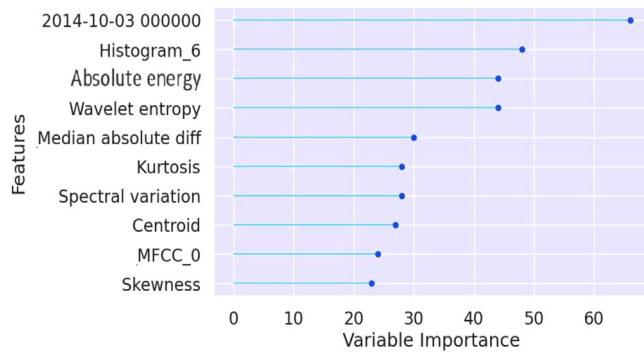


FIGURE 18 Feature's importance derived using the KTBoost classifier

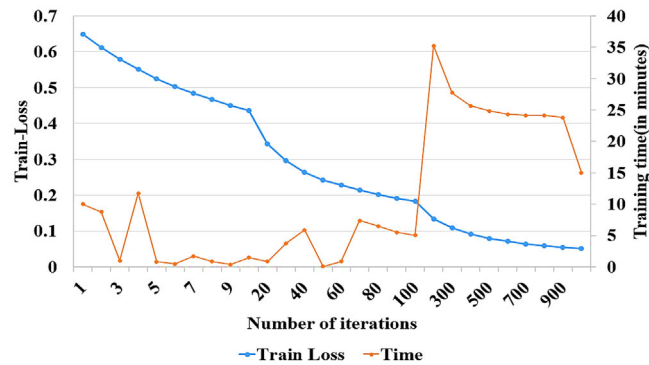


FIGURE 19 The computational time-training loss when the entire feature set of data is provided for model training

Due to the fact that the input training data contains over 1200 features, it is not feasible to display the importance score of each feature in the graph; thus, only the top ten most important features are displayed in Figure 18 together with their importance score. The figure shows that the feature from actual consumption had the highest significance value, followed by statistical features derived from actual consumption. In order to demonstrate the significance of the importance score assigned by the KTBoost model to each feature, the KTBoost model was re-trained to incorporate a much smaller yet essential feature set. Figures 19 and 20 show the computing time required to analyse the entire collection of data features (1071 features) and the 23 most important data features. As can be seen in the mentioned figures, when a smaller number of features set is given, a substantial decrease in computing time is achieved.

In addition to that, Figure 21 depicts the effect of important features on the model's accuracy. The model achieved an accuracy value of 80 percent when just the five most important features were supplied. By increasing the number of important features set from 5 to 23, the model achieved the same accuracy as when trained with all 1071 features. Thus, the conclusion from this can experiment be made that, if the model is retained with the most important features set, the computational resource required can be drastically reduced without violation in accuracy values.

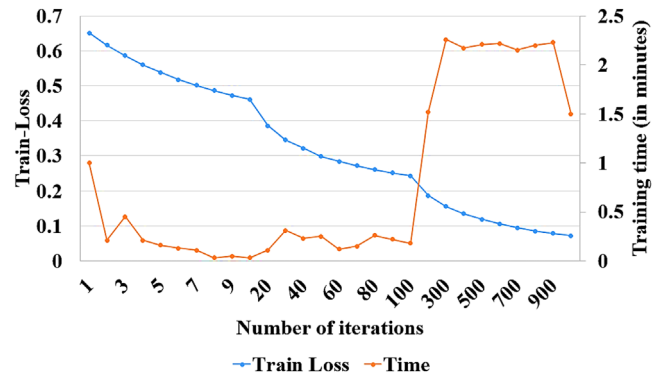


FIGURE 20 The computational time-training loss when the most essential features are provided for model training

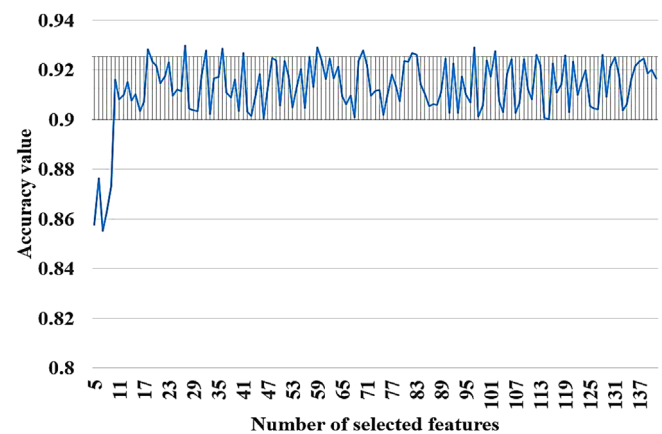


FIGURE 21 Proposed model performance with essential features set

#### 4.6 | Proposed model's comparison against the latest and traditional methods

This section presents a side-by-side comparison of the proposed theft detection framework with a series of well-known traditional machine learning models and the latest bagging and boosting models under an identical feature set. To assess the performance of all studied classifiers, the ten-fold cross-validation method is used in conjunction with the five most commonly used performance measures, namely accuracy, recall, precision, F1-score, Kappa value, and MCC-value.

The proposed framework is sequentially implemented using the Google-Collaboratory (Python 3 Google Compute Engine backend, 12-GB RAM, without GPU-enabled) environment. The comparison's results are summarized in Table 7. As summarized in the table, the proposed approach surpasses all other ML techniques in terms of accuracy, recall, precision,  $F1_{score}$ , Kappa-value, and MCC value, thus evidencing its efficacy and importance. In addition, the proposed model obtained a 93.38% accuracy and recall, the precision of 93.18% and 95%, respectively, which is considerably better than all competing models.



**TABLE 7** Proposed model comparison against latest and traditional ML methods

Model	Accuracy	Recall	Precision	F1-score	Kappa-value	MCC
Proposed model	0.9338	0.9318	0.9508	0.9371	0.8873	0.9077
XGBoost classifier	0.9112	0.9123	0.9012	0.912	0.867	0.875
Extra tree classifier	0.901	0.8921	0.912	0.934	0.854	0.812
SNAP boost algorithm	0.90	0.8912	0.9216	0.9123	0.8412	0.845
lightGBM	0.891	0.8751	0.8631	0.8641	0.8124	0.854
Wide-Deep CNN	0.89	0.812	0.881	0.7921	0.812	0.8213
Gaussian process based boosting	0.885	0.8754	0.8698	0.8412	0.8421	0.7892
Boosted C5.0 algorithm	0.881	0.8541	0.824	0.8121	0.824	0.8245
NGBoost algorithm	0.87	0.861	0.834	0.8251	0.834	0.8964
Random-forest classifier	0.834	0.8123	0.8241	0.8125	0.8453	0.831
SVM - linear Kernel	0.823	0.7601	0.8292	0.7928	0.6042	0.6066
AdaBoost classifier	0.814	0.7562	0.7213	0.745	0.751	0.761
Ridge classifier	0.795	0.7931	0.8584	0.8244	0.6622	0.6641
Quadratic discriminant analysis	0.721	0.2251	0.8911	0.3594	0.1976	0.2974
Logistic regression	0.712	0.8063	0.8482	0.8267	0.6619	0.6627
Linear discriminant analysis	0.698	0.7929	0.8583	0.8243	0.662	0.6639
K neighbour's classifier	0.587	0.6412	0.7606	0.8284	0.6233	0.6356
Naive Bayes	0.54	0.3478	0.6261	0.4472	0.1401	0.1563

## 5 | CONCLUSION

This study presented a novel sequentially executed data-driven approach for identifying electric fraud in a smart meter dataset. The raw smart meter data often contains several null and irregular values mostly due to the malfunction of equipment, poor network, or device storage-related issues. Since most machine learning classifiers cannot process the null values present in the data; therefore, this study estimated missing values using an ensemble machine learning-based predictive modelling technique called XGBoost. Afterward, the robust-SMOTE algorithm was used to balance the class distribution in the acquired data. By considering all regions of minority samples in the dataset, the robust-SMOTE technique produces the minority class samples that are less prone to overfitting and noisy sample generation. Once a balanced dataset is obtained, a set of statistical, temporal, and spectral features were extracted from it. These additional features aid the ML-classifier in understanding the underlying complicated data patterns contained in the data. Finally, in order to effectively classify the data into “Honest” and “Fraudster” consumers, the Jaya optimized KTBoost classifier was used. The Jaya-KTBoost technique combines kernel boosting and tree boosting with its hyperparameters are tuned by utilizing the intelligence of the Jaya algorithm. The proposed model attained an accuracy of 93.38%, precision of 95%, and recall of 93.11%, which are significantly higher than all compared methods.

## FUNDING INFORMATION

This work was supported by the Fundamental Research Grant Scheme under Grant R.J130000.7851.5F062 through the Ministry of Higher Education, Malaysia.

## CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study is publicly available at: <https://github.com/henryRDlab/ElectricityTheftDetection>.

## ACKNOWLEDGEMENT

This work was supported by the Fundamental Research Grant Scheme under Grant R.J130000.7851.5F062 through the Ministry of Higher Education, Malaysia.

## ORCID

Saddam Hussain  <https://orcid.org/0000-0003-1466-7773>

## REFERENCES

- Jenkins, N., Long, C., Wu, J.: An overview of the smart grid in Great Britain. *Engineering* 1(4), 413–421 (2015)
- Liu, Y., Yu, Y., Gao, N., Wu, F.: A grid as smart as the internet. *Engineering* 6(7), 778–788 (2020)

3. Wang, Y., Chen, Q., Kang, C., Xia, Q.: Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE Trans. Smart Grid* 7(5), 2437–2447 (2016)
4. Quilumba, F.L., Lee, W.-J., Huang, H., Wang, D.Y., Szabados, R.L.: Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Trans. Smart Grid* 6(2), 911–918 (2015). <https://doi.org/10.1109/tsg.2014.2364233>
5. Sun, M., Wang, Y., Strbac, G., Kang, C.: Probabilistic peak load estimation in smart cities using smart meter data. *IEEE Trans. Ind. Electron.* 66(2), 1608–1618 (2018)
6. Samadi, P., Mohsenian-Rad, A.-H., Schober, R., Wong, V.W.S., Jatskevich, J.: Optimal real-time pricing algorithm based on utility maximization for smart grid. In: 2010 First IEEE International Conference on Smart Grid Communications. Gaithersburg, MD, pp. 415–420 (2010)
7. Sengan, S., Subramaniaswamy, V., Indragandhi, V., Velayutham, P., Ravi, L.: Detection of false data cyber-attacks for the assessment of security in smart grid using deep learning. *Comput. Electr. Eng.* 93, 107211 (2021)
8. Wei, D., Lu, Y., Jafari, M., Skare, P.M., Rohde, K.: Protecting smart grid automation systems against cyberattacks. *IEEE Trans. Smart Grid* 2(4), 782–795 (2011)
9. Shipworth, D., Fell, M.J., Elam, S.: Response to vulnerability and resistance in the United Kingdom's smart meter transition. *Energy Policy* 124, 418–420 (2019)
10. Messinis, G.M., Hatziaargyriou, N.D.: Review of non-technical loss detection methods. *Electr. Power Syst. Res.* 158, 250–266 (2018). <https://doi.org/10.1016/j.epsr.2018.01.005>
11. Smith, T.B.: Electricity theft: A comparative analysis. *Energy Policy* 32(18), 2067–2076 (2004)
12. Nagi, J., Yap, K.S., Tiong, S.K., Ahmed, S.K., Mohamad, M.: Nontechnical loss detection for metered customers in power utility using support vector machines. *IEEE Trans. Power Delivery* 25(2), 1162–1171 (2009)
13. Saeed, M.S., Mustafa, M.W., Sheikh, U.U., Jumani, T.A., Mirjat, N.H.: Ensemble bagged tree based classification for reducing non-technical losses in multian electric power company of Pakistan. *Electronics* 8(8), 860 (2019)
14. Liu, Y., Shiyan, H.: Cyberthreat analysis and detection for energy theft in social networking of smart homes. *IEEE Transactions on Computational Social Systems* 2.4, 148–158 (2015)
15. Spirić, J.V., Dočić, M.B., Stanković, S.S.: Fraud detection in registered electricity time series. *Int. J. Electr. Power Energy Syst.* 71, 42–50 (2015)
16. Villar-Rodríguez, E., Del Ser, J., Oregi, I., Bilbao, M.N., Gil-Lopez, S.: Detection of non-technical losses in smart meter data based on load curve profiling and time series analysis. *Energy* 137, 118–128 (2017)
17. Yip, S.-C., Wong, K., Hew, W.-P., Gan, M.-T., Phan, R.C.W., Tan, S.-W.: Detection of energy theft and defective smart meters in smart grids using linear regression. *Int. J. Electr. Power Energy Syst.* 91, 230–240 (2017)
18. Lin, C.-H., Chen, S.-J., Kuo, C.-L., Chen, J.-L.: Non-cooperative game model applied to an advanced metering infrastructure for non-technical loss screening in micro-distribution systems. *IEEE Trans. Smart Grid* 5(5), 2468–2469 (2014)
19. Wei, L., Sundararajan, A., Sarwat, A.L., Biswas, S., Ibrahim, E.: A distributed intelligent framework for electricity theft detection using benford's law and stackelberg game. In: 2017 Resilience Week (RWS). Wilmington, DE, pp. 5–11 (2017)
20. Guerrero Alonso, J.I., León de Mora, C., Monedero Goicoechea, I.L., Biscarri Triviño, F., Biscarri Triviño, J.: Improving knowledge-based systems with statistical techniques, text mining, and neural networks for non-technical loss detection. *Knowledge-Based Syst.* 71, 376–388 (2014)
21. Meira, J.A., et al.: Distilling provider-independent data for general detection of non-technical losses. In: 2017 IEEE Power and Energy Conference at Illinois (PECI). Champaign, IL, pp. 1–5 (2017)
22. Avila, N.F., Figueroa, G., Chu, C.-C.: NTL detection in electric distribution systems using the maximal overlap discrete wavelet-packet transform and random undersampling boosting. *IEEE Trans. Power Syst.* 33(6), 7171–7180 (2018)
23. Hussain, S., Mustafa, M.W., Jumani, T.A., Baloch, S.K., Saeed, M.S.: A novel unsupervised feature-based approach for electricity theft detection using robust PCA and outlier removal clustering algorithm. *Int. Trans. Electr. Energy Syst.* 30(11), e12572 (2020)
24. Salman Saeed, M., et al.: An efficient boosted C5.0 decision-tree-based classification approach for detecting non-technical losses in power utilities. *Energies* 13(12), 3242 (2020). <https://doi.org/10.3390/en13123242>
25. Gunturi, S.K., Sarkar, D.: Ensemble machine learning models for the detection of energy theft. *Electr. Power Syst. Res.* 192, 106904 (2020). <https://doi.org/10.1016/j.epsr.2020.106904>
26. Fei, K., Li, Q., Zhu, C.: Non-technical losses detection using missing values' pattern and neural architecture search. *Int. J. Electr. Power Energy Syst.* 134, 107410 (2022)
27. Elhassan, A., Abu-Soud, S.M., Alghanim, F., Salameh, W.: ILA4: Overcoming missing values in machine learning datasets—An inductive learning approach. *J. King Saud Univ.-Comput. Inf. Sci.* (2021)
28. Hastie, T., Tibshirani, R., Friedman, J.: *The Elements of Statistical Learning*. Springer-Verlag, New York (2001)
29. Noor, N.M., Al Bakri Abdullah, M.M., Yahaya, A.S., Ramli, N.A.: Comparison of linear interpolation method and mean method to replace the missing values in environmental data set. *Trans. Tech. Publ.* 803, 278–281 (2015)
30. Ansley, C.F., Kohn, R.: On the estimation of ARIMA models with missing values. In: *Time Series Analysis of Irregularly Observed Data*. pp. 9–37. Springer, New York (1984)
31. Troyanskaya, O., et al.: Missing value estimation methods for DNA microarrays. *Bioinformatics* 17(6), 520–525 (2001). <https://doi.org/10.1093/bioinformatics/17.6.520>
32. Rahman, M.G., Islam, M.Z.: Missing value imputation using a fuzzy clustering-based EM approach. *Knowledge Inf. Syst.* 46(2), 389–422 (2016)
33. Wang, X., Li, A., Jiang, Z., Feng, H.: Missing value estimation for DNA microarray gene expression data by Support Vector Regression imputation and orthogonal coding scheme. *BMC Bioinf.* 7(1), 1–10 (2006)
34. Tang, F., Ishwaran, H.: Random forest missing data algorithms. *Stat. Anal. Data Mining* 10(6), 363–377 (2017)
35. Luong, P., Nguyen, D., Gupta, S., Rana, S., Venkatesh, S.: Bayesian optimization with missing inputs. *arXiv preprint arXiv:2006.10948* (2020)
36. Hasan, M., Toma, R.N., Nahid, A.-A., Islam, M.M., Kim, J.-M.: Electricity theft detection in smart grid systems: A CNN-LSTM based approach. *Energies* 12(17), 3310 (2019)
37. Gunturi, S.K., Sarkar, D.: Ensemble machine learning models for the detection of energy theft. *Electric Power Systems Research* 192, 106904 (2021). <https://doi.org/10.1016/j.epsr.2020.106904>
38. Buzau, M.-M., Tejedor-Aguilera, J., Cruz-Romero, P., Gomez-Exposito, A.: Detection of non-technical losses using smart meter data and supervised learning. *IEEE Trans. Smart Grid* 10, 2661–2670 (2018)
39. Javaid, N., Jan, N., Javed, M.U.: An adaptive synthesis to handle imbalanced big data with deep siamese network for electricity theft detection in smart grids. *J. Parallel Distrib. Comput.* 153, 44–52 (2021)
40. Punmiya, R., Choe, S.: Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Trans. Smart Grid* 10(2), 2326–2329 (2019). <https://doi.org/10.1109/tsg.2019.2892595>
41. Joker, P., Arianpoo, N., Leung, V.C.M.: Electricity theft detection in AMI using customers' consumption patterns. *IEEE Trans. Smart Grid* 7(1), 216–226 (2015)
42. Oprea, S.-V., Băra, A.: Machine learning classification algorithms and anomaly detection in conventional meters and Tunisian electricity consumption large datasets. *Comput. Electr. Eng.* 94, 107329 (2021)
43. Nizar, A.H., Dong, Z.Y., Zhao, J.H., Zhang, P.: A data mining based NTL analysis method. *IEEE*, pp. 1–8 %@ 142441296X (2007)
44. Nagi, J., Yap, K.S., Tiong, S.K., Ahmed, S.K., Mohammad, A.M.: Detection of abnormalities and electricity theft using genetic support vector machines. In: TENCON 2008 - 2008 IEEE Region 10 Conference. Hyderabad, India, pp. 1–6 (2008)
45. Nizar, A.H., Dong, Z.Y.: Identification and detection of electricity customer behaviour irregularities. *IEEE/PES Power Systems Conference and Exposition*, pp. 1–10 (March, 2009)
46. Nagi, J., Yap, K.S., Tiong, S.K., Ahmed, S.K., Mohamad, M.: Nontechnical loss detection for metered customers in power utility using support vector

- machines. *IEEE Trans. Power Delivery* 25(2), 1162–1171 (2010). <https://doi.org/10.1109/tpwrld.2009.2030890>
47. Ramos, C.C.O., de Sousa, A.N., Papa, J.P., Falcao, A.X.: A new approach for nontechnical losses detection based on optimum-path forest. *IEEE Trans. Power Syst.* 26(1), 181–189 (2010)
  48. Ramos, C.C.O., Souza, A.N., Chiachia, G., Falcão, A.X., Papa, J.P.: A novel algorithm for feature selection using harmony search and its application for non-technical losses detection. *Comput. Electr. Eng.* 37(6), 886–894 (2011)
  49. León, C., Biscarri, F., Monedero, I., Guerrero, J.I., Biscarri, J., Millán, R.: Integrated expert system applied to the analysis of non-technical losses in power utilities. *Expert Syst. Appl.* 38(8), 10274–10285 (2011)
  50. Faria, L.T., Melo, J.D., Padilha-Feltrin, A.: Spatial-temporal estimation for nontechnical losses. *IEEE Trans. Power Delivery* 31(1), 362–369 (2015)
  51. Kosut, J.P., Santomauro, F., Jorysz, A., Fernández, A., Lecumberry, F., Rodríguez, F.: Abnormal consumption analysis for fraud detection: UTE-UDELAR joint efforts IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM). *IEEE*, pp. 887–892 (2015)
  52. Jokar, P., Arianpoo, N., Leung, V.C.M.: Electricity theft detection in AMI using customers' consumption patterns. *IEEE Trans. Smart Grid* 7(1), 216–226 (2016). <https://doi.org/10.1109/tsg.2015.2425222>
  53. Nallathambi, S., Ramasamy, K.: Prediction of electricity consumption based on DT and RF: An application on USA country power consumption. *IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)*, pp. 1–7 (2017)
  54. Zheng, Z., Yang, Y., Niu, X., Dai, H.-N., Zhou, Y.: Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Trans. Ind. Inf.* 14(4), 1606–1615 (2018). <https://doi.org/10.1109/tii.2017.2785963>
  55. Blazakis, K.V., Kapetanakis, T.N., Stavrakakis, G.S.: Effective electricity theft detection in power distribution grids using an adaptive neuro fuzzy inference system. *Energies* 13(12), 3110 (2020)
  56. Qu, Z., Li, H., Wang, Y., Zhang, J., Abu-Siada, A., Yao, Y.: Detection of electricity theft behavior based on improved synthetic minority oversampling technique and random forest classifier. *Energies* 13(8), 2039 (2020)
  57. Lin, G., Feng, X., Guo, W., Cui, X., Liu, S., Jin, W., Lin, Z., Ding, Y.: Electricity theft detection based on stacked autoencoder and the undersampling and resampling based random forest algorithm. *IEEE Access* 9, 124044–124058 (2021)
  58. Ullah, A., Munawar, S., Asif, M., Kabir, B., Javaid, N.: Synthetic theft attacks implementation for data balancing and a gated recurrent unit based electricity theft detection in smart grids. In: *Complex, Intelligent and Software Intensive Systems. CISIS 2021. Lecture Notes in Networks and Systems*. Vol. 278, pp. 395–405. Springer, Cham (2021)
  59. Cheng, G., Zhang, Z., Li, Q., Li, Y., Jin, W.: Energy theft detection in an edge data center using deep learning. *Math. Prob. Eng.* 2021, 1–12 (2021)
  60. Umer, M., et al.: Scientific papers citation analysis using textual features and SMOTE resampling techniques. *Pattern Recognit. Lett.* 150, 250–257 (2021)
  61. Chen, B., Xia, S., Chen, Z., Wang, B., Wang, G.: RSMOTE: A self-adaptive robust SMOTE for imbalanced problems with label noise. *Inf. Sci.* 553, 397–428 (2021)
  62. Heaton, J.: An empirical analysis of feature engineering for predictive modeling. In: *SoutheastCon 2016*. Norfolk, VA, pp. 1–6 (2016)
  63. Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E.B., Turaga, D.S.: Learning feature engineering for classification. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*. Melbourne, Australia, pp. 2529–2535 (2017)
  64. Christ, M.: TSFRESH - A python package documentation - Time series feature extraction on basis of scalable hypothesis tests (2019). Accessed on: 24 December, 2021. [https://tsfresh.readthedocs.io/\\_/downloads/en/v0.12.0/pdf/](https://tsfresh.readthedocs.io/_/downloads/en/v0.12.0/pdf/)
  65. González, S., García, S., Del Ser, J., Rokach, L., Herrera, F.: A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. *Inf. Fusion* 64, 205–237 (2020)
  66. Sigrist, F.: KTBoost: Combined kernel and tree boosting. *Neural Process. Lett.* 53(2), 1147–1160 (2021)
  67. Guerrero, J.I., Monedero, I., Biscarri, F., Biscarri, J., Millan, R., León, C.: Non-technical losses reduction by improving the inspections accuracy in a power utility. *IEEE Trans. Power Syst.* 33(2), 1209–1218 (2017)
  68. Perlich, C., Provost, F., Simonoff, J.: Tree induction vs. logistic regression: A learning-curve analysis. *J. Mach. Learn. Res.* 4, 211–255 (2003)

**How to cite this article:** Hussain, S., Mustafa, M.W., Ateyeh Al-Shqeerat, K.H., et al.: Electric theft detection in advanced metering infrastructure using Jaya optimized combined Kernel-Tree boosting classifier—A novel sequentially executed supervised machine learning approach. *IET Gener. Transm. Distrib.* 16, 1257–1275 (2022). <https://doi.org/10.1049/gtd2.12386>