IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 November 2022, accepted 15 December 2022, date of publication 20 December 2022,
date of current version 28 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3230952

RESEARCH ARTICLE

Adaptive Data Balancing Method Using Stacking Ensemble Model and Its Application to Non-Technical Loss Detection in Smart Grids

ASHRAF ULLAH[1], NADEEM JAVAID[1], (Senior Member, IEEE), MUHAMMAD UMAR JAVED[1], (Graduate Student Member, IEEE), PAMIR[1], (Graduate Student Member, IEEE), BYUNG-SEO KIM[2], (Senior Member, IEEE), AND SAEED ALI BAHAJ[3]

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[2]Department of Computer and Information Communications Engineering, Hongik University, Sejong 30016, South Korea
[3]Department of Management Information Systems, College of Business Administration, Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia

Corresponding authors: Nadeem Javaid (nadeemjavaidqau@gmail.com) and Byung-Seo Kim (jsnbs@hongik.ac.kr)

This work was supported by the National Research Foundation (NRF), South Korea, under Project BK21 FOUR.

ABSTRACT A stacking ensemble model (SEM) is proposed in this paper to identify non-technical losses. Three layers make up the proposed model. Data pre-processing is performed at the first layer, where issues of data imbalance, missing values, and data normalization are dealt with. Min-max and a simple imputer are used to handle data normalization and missing values, respectively. Besides, ADASYN and TomekLink are used in a combined form to address the problem of data imbalance. The second layer employs three different machine learning models. The models, also referred to as base classifiers, used at the second layer in the proposed SEM include the following classifiers: random forest (RF), extra tree (ET), and extreme gradient boosting (XGBoost). To accomplish the final classification using the ridge classifier, the output of the basic classifiers is ensembled at the third layer. The ridge classifier is also regarded as the meta classifier. Furthermore, the training and testing of the suggested model is aided by real-time data from the smart grid corporation of China (SGCC). The proposed model's performance is validated by multiple simulations using various performance indicators and is found to surpass the standalone classifiers in terms of ETD.

INDEX TERMS ADASYN, deep learning, non-technical losses, SGCC, stacking model, smart grids, TomekLink.

I. INTRODUCTION

The power grids comprise power generators, power distributors and power transmission lines, and form complicated designs. In such grids, detecting losses on a real-time basis is extremely challenging due to the lack of an intelligent system. The one-way communication between the grid and the end users is another issue that these grids face. It prevents the traditional grids from providing electricity to end customers on demand in real-time. These issues harm the performance of the power grids and reduce their lifespan. The incorporation of the advanced metering infrastructure (AMI) makes smart grids a reality. The bidirectional communication between the grid and the end users is made possible via the smart grid. However, they face energy losses, broadly grouped into two categories: technical and non-technical. The losses increased from 11% to 16% from 1980 to 2000 [1]. In the USA, the losses are 6% while in Russia, Brazil and India the losses are 6%, 10% and 16% of the whole energy production, respectively [2]. Technical losses (TLs) are brought on by faulty transmission lines or transformers. Contrarily, non-technical losses (NTLs) are brought about by issues with metres, improper metre installation, theft of electricity, etc., [3], [4]. Through effective electricity theft

The associate editor coordinating the review of this manuscript and approving it for publication was Salvatore Favuzza.

detection (ETD), our main objective is to find NTLs. It is based on several approaches, which are broadly classified into the following three categories [5].

1. State-based methodologies: these methodologies measure users' electricity consumption using various hardware components, such as sensors and radio frequency identification tags. These are also known as the hardware-based methodologies. Such methodologies have more robust ETD performance. However, they face maintenance issues.

2. Game-based methodologies: in such methodologies, a game is played between electricity consumers and attackers to profit both entities [6]. As such methodologies are based on assumptions, they do not produce efficient ETD results.

3. Machine learning methodologies: such methodologies analyze the users' electricity consumption trends using variety of machine learning approaches. These methodologies give the best ETD performance. Such methodologies are also used in various fields of life like healthcare, education, transport, etc.

In the ETD, the number of dishonest users is minimal. However, the number of honest users is large. Such a situation is referred to as class imbalance, and is not good in the context of ETD. It is because it gets challenging for the classifier to make a fair categorization in such conditions. In [7] and [8], the authors develop several artificial intelligence (AI) based strategies for analyzing users' electricity usage patterns to identify suspected customers. However, the desired results are not obtained. Besides, the literature is found to have the following limitations.

- Class imbalance: it is one of the leading issues in terms of ETD. During classification, the classifier tends towards the majority class instances, due to this issue. In such situations, the accuracy is misleading as the minority class is confused for the majority class to maximize the accuracy of a classifier. To overcome this issue, various oversampling, undersampling and hybrid techniques are proposed [9]. To solve this issue, we put forward the ADASYN and TomekLink based hybrid technique in the proposed model.
- High dimensional problem: the selection of relevant features is also one of the leading problems in ETD. Various machine and deep learning approaches are used to solve this issue [10]. We propose machine learning-based methods to solve this issue.
- High false positive rate (FPR): high FPR is also one of the leading issues in the context of ETD. It occurs due to misclassification [11]. The proposed work resolves the issue by the combination of ADASYN and TomekLink.
- Overfitting: when a classifier is trained to a greater extent or with small amount of data volume compared to the input dimension, it does not perform well. We proposed the ridge-based classifier in the proposed work that includes the penalty constraint. The objective is to solve the overfitting problem [10], [11].
- For both classification and selection, researchers mostly use various convolutional deep neural networks (DNNs), which are flavors of original convolutional neural network (CNN) along with Alexnet. These strategies mostly use different activation functions like sigmoid, tanh, relu, etc. These activation functions face both vanishing gradient and exploding gradient issues. As a result, the learning process is affected to a greater extent [12].

We proposed a viable approach for analyzing massive data for the ETD in this research work. The primary goal is to efficiently distinguish between honest and dishonest users based on their electricity usage patterns. The following are the paper's main contributions.

- A hybrid technique based on ADASYN and TomekLink is utilized for tackling the problem of class imbalance.
- A stacking ensemble model (SEM) is proposed for detecting the NTLs in the underlying work. The model employs three classifiers at level-0 and one classifier at level-1.
- The proposed model's performance is validated using different performance metrics.

The following is the organization of the paper. Section II discusses the existing models that have been employed for ETD. Sections III and IV define the problem statement and presents the proposed system model. Section V presents the problem formulation. Section VI provides the model evaluation. The simulation results are discussed in Section VII. Section VIII presents the conclusion.

II. LITERATURE SURVEY

This section presents the literature review. The current works are broadly categorized into three groups.

A. METAHEURISTIC TECHNIQUES FOR PARAMETER TUNING AND FEATURE SELECTION

In this section, numerous possible metaheuristic strategies that have been used in the literature for parameter tuning and feature selection are discussed. The basic goal is to achieve the most satisfactory possible convergence. For ETD, CNN-gated recurrent unit (GRU)-particle swarm optimization (PSO) based deep hybrid model (HDM) is proposed by the authors of [5]. In the proposed HDM, CNN chooses the most relevant features while GRU is used to classify them. Besides, PSO is used to fine-tune the GRU's parameters. The proposed HDM's goal is to increase the model's accuracy and make it more resilient against outliers. The authors in [13] use the long short term memory (LSTM) to deal with data dimensionality. Random undersampling (RUS)-Boost method is utilized for classification. It is a hybrid of RUS and AdaBoost algorithm. Undersampling is done by RUS, while boosting is done by AdaBoost. To solve the binary classification problem in ETD, the AdaBoost parameters are sent to a metaheuristic approach. The authors proposed the HDM in [14]. The visual geometry group (VGG-16) is utilized in the proposed HDM to choose the target features. The binary classifications are done using

firefly algorithm (FA) based XGBoost. FA is used for parameter tuning. The goal is to describe the difference between fraudulent and innocent users. The black hole algorithm (BHA) method is proposed in [15] to select the most representative features. The paper's primary goal is to pinpoint the irregular consumption patterns and identify commercial losses.

B. OVERSAMPLING AND UNDERSAMPLING TECHNIQUES TO HANDLE DATA IMBALANCE

In [16], TomekLink borderline synthetic minority oversampling technique with support vector machine (TBSSVM) is proposed to tackle the data imbalance issue. While the combined model of temporal correlation network (TCN) and enhanced multi-layer perceptron (EMLP) is proposed for classification. Rusboost is combined with the maximal overlap discrete wavelet-packet transform (MODWPT) in [17] for feature engineering. The validation results demonstrate the accuracy of the model proposed in this work. The authors of [18] employ the CNN model to distinguish between normal and abnormal electricity usage patterns. The simulation results demonstrate that the model perform better on random oversampling (ROS) than other sampling methods. The authors in [9] develop an ensemble machine learning approach for ETD and to combat outliers. Synthetic minority oversampling technique (SMOTE) is used to address the problem of data imbalance. The results demonstrate that other benchmark models are less accurate and resilient than the proposed model.

C. FEATURE ENGINEERING TO MAKE THE MODEL ROBUST AGAINST ATTACKS

The authors in [11] use the gradient boosting theft detector (GBTD). The proposed strategy defends against multiple attacks by combining the boosting techniques: XGBoost, CatBoost, and LightBoost. To reduce the FPR, various stochastic features like mean, standard deviation, etc., are used. The gradient boosting classier is used for ETD by the authors of [19]. Genetic algorithm (GA) is used to build new features from existing ones. Various attacks are induced for the model testing. The results demonstrates that the suggested model is more reliable and accurate than another benchmark models.

III. PROBLEM STATEMENT

The power grid is being harmed by electricity theft [20]. NTLs are the primary cause of it. In the imbalanced dataset [9], the number of trustworthy users is more than the number of dishonest users. Maintaining a balance between honest and fraudulent consumers' consumption is challenging when using such a dataset. In the literature, oversampling and undersampling-based class imbalance approaches are used [9], [18] as the foundation for machine learning and deep learning-based algorithms. The proposed techniques have poor learning and generalization capabilities since they either generate synthetic data or randomly remove data from the majority class. Consequently, high FPR is achieved [11]. These issues further lead to less accurate results and depreciated robustness when dealing with outliers. Therefore, to address all the mentioned issues, SEM-based stacking model is proposed.

TABLE 1. Description of SGCC dataset.

Attribute(s)	Value(s)
Electricity consumption time window	2014/01/01– 2016/10/31
Total number of customers	42372
Number of normal (honest) users	38757
Number of fraudulent (theft) users	3615

IV. PROPOSED MODEL

The proposed method is composed of a data pre-processing and stacking model. The stacking model further comprises four different classifiers, employed at levels 0 and 1. The following subsections discuss the proposed system model in more depth.

A. PRE-PROCESSING OF DATA

In this stage, the data imbalance issue is handled using ADASYN, while data normalization is perfomed and the missing values are imputed using min-max and interpolation techniques. The details of the steps involved in data pre-processing are discussed below. Table 1 provides a description of the SGCC dataset [21].

1) HANDLING THE DATA IMBALANCE ISSUE

The imbalance data problem is tackled via different strategies in the ETD context. These strategies are broadly classified into three categories: data level, algorithmic and hybrid. The objective of data-level strategies is to improve sample gathering involved in data distribution. In algorithmic strategies, the work is done on the algorithm rather than data. In the hybrid strategies [22], the positive aspects of data level and algorithmic strategies are combined. The researchers used numerous data sampling algorithms to balance the data in the literature. To manage the imbalance data, SMOTE is used [9], [12]. This strategy replicates instances in the minority class at random, and is prone to overfitting [20]. Besides, RUS is also employed in the data balancing process. The issue with RUS is that it causes information loss [18]. To deal with this issue, near miss (NM) technique is used [23], [24]. Rather than data level or hybrid strategies, our focus is on algorithmic strategies. In this paper, we employed a hybrid method based on ADASYN [14] and TomekLink [16]. As illustrated in Figure 1, ADASYN is an oversampling technique. Whereas, TomekLink is an undersampling technique, as shown in Figure 2. The mathematical representation of ADASYN is given in the following equations.

$$d = \frac{m_s}{m_i} \tag{1}$$

Equation 1 describes the ratio between the majority and minority classes. m_i represents the minority class instances

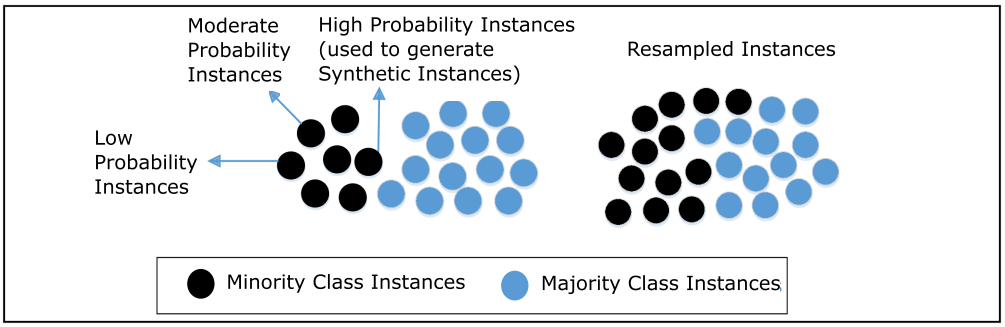


FIGURE 1. Workflow of ADASYN: resampling of imbalanced class instances.

while m_s represents the majority class instances.

$$G = (m_s - m_i) * \beta \tag{2}$$

Equation 2 describes the number of synthetic observations between two classes that are balanced using β, keeping its value equal to 1.

$$r_i = \triangle_i / K, \quad i = 1, 2, \ldots, m \tag{3}$$

In Equation 3, the synthesized samples are represented by $\triangle_i$ while i denote the number of neighbors that belong to the majority class. Suppose, K is selected to be five. So for a particular class, two out of five observations are from the majority class, which are represented by the $\triangle_i$. As a result, the r_i for the specific minority class is 0.4. r_i is normalized using the density distribution $\hat{r}_i$ via Equation 4.

$$\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i \tag{4}$$

Equation 5 determines the number of points synthetically generated for any individual minority point.

$$g_i = \hat{r}_i * G \tag{5}$$

ADASYN oversampling technique intelligently creates synthetic samples for any instance of the probability distribution. As seen in Figure 1, instances with high probability generate large number of synthetic instances, instances with moderate probability create moderate number of instances while the instances with the lowest probability create the least number of instances. Unlike ADASYN, the TomekLink technique [16] removes unneeded instances from the majority class, creating a balance between the majority and the minority classes, as demonstrated in Figure 2. TomekLink is an undersampling technique that makes pairs of data points (m_n, m_j). The m_n denotes the minority class instance while m_j represents the majority class instance. The pair (m_n,m_j) forms a TomekLink in the case of the condition being unsatisfied by sample x_k, given as $d(m_n, x_k) < d(m_n, m_j)$. Also, given as $d(m_j, x_k) < d(m_n, m_j)$. By doing so, the removal of the majority class samples placed near the minority class

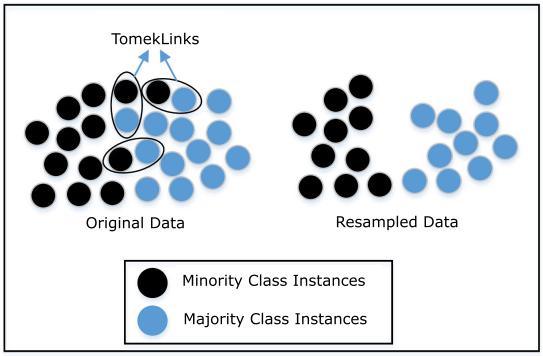


FIGURE 2. Workflow of TomekLink.

samples happens, which leads to data balancing. Besides, both ADASYN and TomkeLink are combined in a hybrid approach through a pipeline technique [25].

2) DATA NORMALIZATION

The neural network model is very sensitive to data, so every significant or minor change in the data affects the learning process. To tackle this issue, normalization proves to be the most effective method. Equation 6 presents the mathematical formulation of min-max normalization [12].

$$Z = \frac{B - min(B)}{max(B) - min(B)} \tag{6}$$

The consumer's energy consumption pattern is described by B. The difference between the upper and lower bounds of min and max functions represents the amount of electricity consumed. As Equation 6 is the ratio, its value lies in the range 0 to 1. Thus, the normalized value of the user's consumed energy lies between 0 and 1.

3) HANDLING THE MISSING VALUES

Generally, each dataset contains missing values or NaN values, which are to be removed before classification. Usually, for filling the NaN values, a simple imputer technique is employed. In this paper, we also used the simple imputer technique to handle the missing values [26].

4) REMOVING OUTLIERS

The outliers are the values that are different from other values of the dataset under observation. The outliers affect the performance of the classifier. Hence, they are removed during classification with the aid of three-sigma rule of thumb [10].

B. STACKING MODEL

After pre-processing, a stacking model is used to classify the data. The stacking model works with two types of classifiers: base classifier and meta classifier. The base classifiers include heterogeneous classifiers that are trained individually on the dataset to perform classification before being handed to the meta-classifier for final classification, as shown in Figure 4. The overall goal of the stacking model is to increase classification accuracy.

The proposed SEM employs three classifiers at level-0 and one classifier at level-1. RF, ET and XGBoost are employed at level-0 as base-learners while ridge classifier is employed at level-1 as a meta-learner. Various machine learning models have been used in the ETD to perform classification like support vector machine (SVM), logistic regression (LR), etc. The problems with these classifiers are overfitting and limited generalization capability [13]. Hence, the overfitting issue is resolved with the help of ridge regression used in combination with the regularization parameters. The correlation between each base learner's prediction result is also dealt with using the same combination.

In ETD, various techniques are used to distinguish the fraudulent and honest users. However, accuracy still remains the main concern. The stack-based machine learning model improves the performance due to the efficient performance of the base models, which are very skillful in solving the problem at hand in different ways [9], [25], [32], [33].

1) BASE CLASSIFIERS

The base classifiers used in the proposed stacking model are RF, ET and XGBoost. The RF and ET are bagging techniques while XGBoost is a boosting technique. In the bagging technique, several weak learners (decision tree models) are combined to estimate the final standardized output. The working of bagging technique is provided in Algorithm 1 [27].

Using the replacement technique, random bootstraps (small samples) are generated in the bagging process. The working of the bootstrap technique is given in Algorithm 2 [10], [27]. The way to better utilize and comprehend the entire dataset is demonstrated in Figure 5.

The boosting is an ensemble technique. In the boosting technique, various weak learners are combined into a single strong learner. The primary difference between bagging and boosting is that in bagging, the weak learners perform in parallel. While in boosting, the weak learners perform sequentially. The working of boosting is given in Algorithm 3 [27].

Algorithm 1 Bagging Technique Algorithm

1: **Initialization**
2: **Input**: Dataset Z = $\{z_1, z_2, \ldots, z_N\}$, with $z_i = (x_i, y_i)$, where $x_i \in$ X and $y_i \in \{0,1\}$, B, number of bootstrap samples.
3: **Output:** Classifier H: X $\rightarrow \{0,1\}$, the final classifier
4: **for** b = 1 to B **do**
5: Draw, with replacement, N samples from Z, obtaining the b-th bootstrap sample Z^*
6: From each bootstrap sample Z^*b, learn classifier H_b.
7: **end for**
8: Produce the final classifier as a majority vote of $H_1, \ldots, H_B$ that is,H(x) = sign($\sum_{b=1}^{B} H_b$(x))
9: **End**

Algorithm 2 Bootstrap Technique Algorithm

1: **Initialization**
2: **Input:** Size-N sample Z $=\{z_1, z_2, \ldots, z_N\}$ of a (potentially infinite) population P.B, number of bootstrap samples
3: **Output**: Estimate $\hat{T}(P)$ of the population statistic
4: **for** b = 1 to B **do**
5: Draw, with replacement, N samples from Z, obtaining the b-th bootstrap sample Z_b^*
6: Compute, for each sample Z_b^*, the estimate of the statistic $\hat{T}(Z_b^*)$
7: **end for**
8: Compute the bootstrap estimate, $\hat{T}(P)$, as the average of $\hat{T}(Z_1^*, \ldots, \hat{T}(Z_B^*)$
9: Compute the accuracy of the estimate, using, e.g., the variance of $\hat{T}(Z_1^*, \ldots, \hat{T}(Z_B^*)$
10: **End**

Algorithm 3 Boosting Technique Algorithm

1: **Initialization**
2: **Input:** Dataset Z = $\{z_1, z_2, \ldots, z_N\}$, with $z_i = (x_i, y_i)$, where $x_i \in$ X and $y_i \in \{0,1\}$
3: **Output:** Classifier H: X $=\{0,1\}$
4: Randomly select, without replacement, L1 < N samples from Z to obtain Z_1^*
5: Run the weak learner on Z_1^*, yielding classifier H1
6: Select L2 < N samples from Z, with half of the samples misclassified by H1, to obtain Z_2^*.
7: Run the weak learner on Z_2^*, yielding classifier H2
8: Select all samples from Z on which H_1 and H_2 disagree, producing Z_3^*
9: Run the weak learner on Z_3^*, yielding classifier H3
10: Produce the final classifier as a majority vote: H(x) = sign($\sum_{b=1}^{3} H_b$(x))
11: **End**

In Algorithm 3, Z_1^*, Z_2^*, Z_3^* represent the samples that are generated with replacement strategy. Besides, H_1, H_2, H_3 represent the weak classifiers. $H(x)$ represents the strong learner or classifier that works on the majority

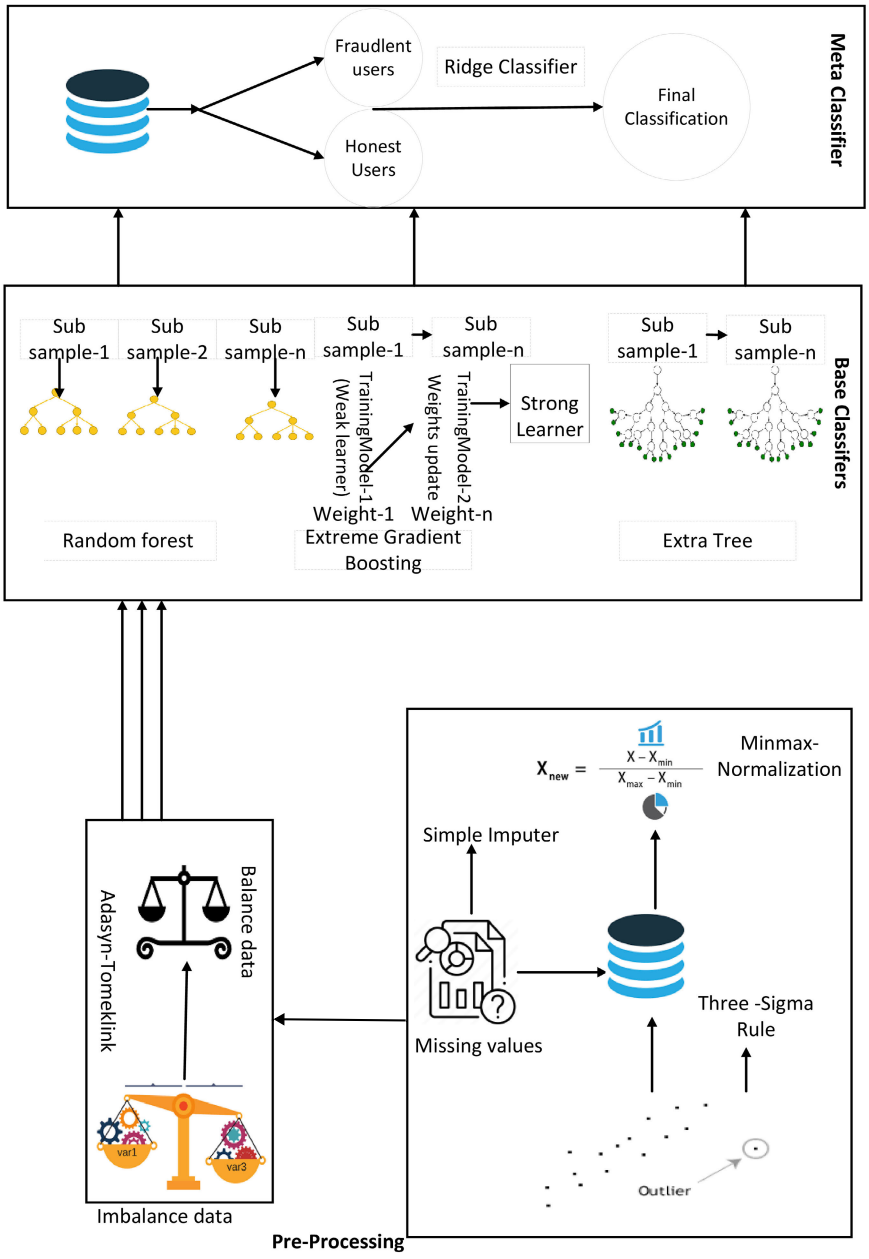


FIGURE 3. Proposed system model.

vote. Overall, boosting is an iterative process in which weights are updated after each iteration and priority is given to misclassified instances. Figure 6 shows the boosting technique. The working of boosting technique is described in Algorithm 3. It is used for performing both classification and regression [28].

The mathematical formulation of the boosting technique is given as follows.

Our goal for each model F is the correct prediction of the values using Equation 7.

$$\hat{y}_i = F(x) \tag{7}$$

The identified mean square error is minimized using Equation 8.

$$\hat{y}_i = \frac{1}{n}\sum_i (\hat{y}_i - y_i)^2 \tag{8}$$

The actual and the predicted values are represented by y_i and $\hat{y}_i$, respectively. n is the number of samples while i represents training over a set of size n instances over the actual values of the output variable y. Besides, to improve the performance of a less efficient model F_m that returns $\hat{y}_i = \bar{y}$, with $\bar{y}$ being the mean value of y, a new estimator $h_m(x)$ is introduced.

$$F_{m+1} = F_m(x_i) + h_m(x_i) = y_i \tag{9}$$

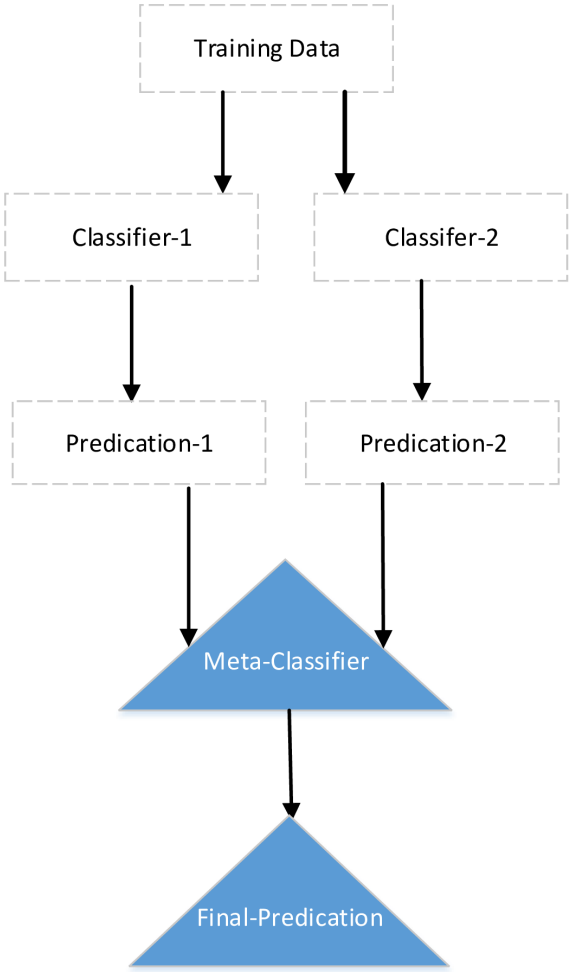


FIGURE 4. Schematic diagram.

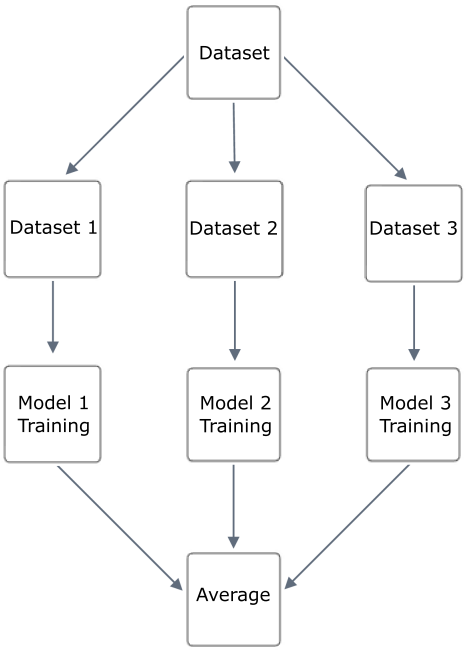


FIGURE 5. Bagging technique.

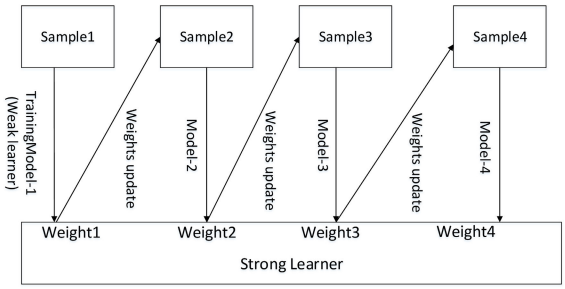


FIGURE 6. Boosting technique.

or it may be

$$h_m(x_i) = y_i - F_m(x_i) \tag{10}$$

The accuracy is improved for each F_{m+1} by attempting to correct the error of the prior $F_{m(x)}$. The major goals are to reduce the loss as in Equations 11 and 12, and increase the accuracy. In this manner, the model is made resilient against outliers.

$$L_{MSE} = \frac{1}{n}\sum_{i}^{n}(y_i - F(x_i)^2) \tag{11}$$

$$-\frac{\partial L_{MSE}}{\partial F} = \frac{2}{n}(y_i - F(x_i)) = \frac{2}{n}h_m(x_i) \tag{12}$$

The base classifiers are discussed in details as follows.

- Random forest
  RF is a bagging technique in which numerous trees are formed during training. The final decisions are made based on the average or votes of each tree, with the goals of improving accuracy and avoiding overfitting. RF may be utilized for both classification and regression as it performs well in the feature selection process. The working of RF is given in Algorithm 4 [10].
  There are three folds of RF [9].
  - It picks data randomly and creates numerous classification trees, making the classifier less susceptible to error or noise.
  - It is adaptable to any dataset and can handle high-dimensional data with ease.
  - Parallel processing allows for rapid training.

  The workflow of RF is as follows [27].
  - Random samples S are bootstrapped from the original dataset using a replacement strategy; $S = S1, S2, \ldots, S_m$, where m is the total number of samples.
  - RF is created by growing a large number of trees on random samples S_m without pruning (strong learner).
  - The best features are selected from the existing features based on the best split.

Algorithm 4 Algorithm of RF

Initialization
Require
i. Training datasets
S = {(x_i, y_j), i = 1, 2, 3 …, m}, (X,Y), ∈ R×R
ii. Testing datasets $x_j \in R^m$
for do 1 to N^{tree} **do**
i. Draw a bootstrap S_d from the original training data
ii. Grow an unpruned tree h_d using data S_d
(a) Randomly select new feature set M_{try} from original feature set e
(b) Select the best features from feature set M_{try} based on Gini indicator on each node
(c) Split until each tree grows to its maximum
end for
Ensure
i. Collection of trees $\{h_d, d = 1, 2, \ldots, N_{tree}\}$
ii. For x_j, the output of a decision tree is $h_d(x_j)$
$f(x_j)$ = majority vote $\{h_d(x_j)\}N_{tree}$
return $f(x_j)$
End

  - The final classification is performed by the majority votes.
  - The *Giniindex* calculates the loss.

- Extreme gradient boosting
  XGBoost is the gradient boosting-based ensemble technique [28]. It is designed to achieve high scalability. The loss in XGBoost is estimated using Equation 13. To reduce the loss, XGBoost uses additive expansion in terms of regularization parameters with an objective function. Furthermore, loss function variation allows for better control of tree complexity.

$$L_{xgb} = \sum_{i}^{N}(y_i, F(x_i)) + \sum_{m=1}^{M}\Omega(h_m) \tag{13}$$

$$\Omega(h_m) = \gamma T + \frac{1}{2}\lambda||w||^2 \tag{14}$$

  T represents the number of leaf nodes, while w represents the output score. The loss is determined using decision tree splitting criteria, which pushes the model towards the pre-pruning approach. The penalty used to control the minimal loss is represented by λ. If λ is large, the loss is large, and the tree becomes simple. If λ is less, the loss causes the tree to be divided into more nodes. Furthermore, these regularization factors decay, resulting in additive expansion with a small step size. Besides, max depth is an important controllable parameter utilized in the XGBoost to manage the tree's complexity, which results in fast training and less consumption of storage space. The random approach makes XGBoost less overfit and quick to train [28]. Furthermore, XGBoost uses various methods to speed up the training process that has nothing to do with the ensemble classifier's accuracy. The major goal is to improve split performance by lowering the complexity. Due to a linear scan over all sorted attributes [28], in the splitting phase, all viable candidates are usually considered and those with the highest gain are chosen. The working of XGBoost is given in Algorithm 5.

Algorithm 5 XGBoost Algorithm

Initialization
Input: Explanatory feature matrix: X; target attribute vector: Y; loss function: l(y,$\hat{y}$); base learner: g(X,μ); number of subtrees: K
Output: Prediction probability
for t=1:K **do**
Initialize $G_o(X_i)$ $argmin_p = \sum_{i=1}^{N} L(y_i, p)$
Compute $\nabla G_t(X)$
Run the new learner function g(X,μ)
Predict the best gradient descent stage $size(p_k) = argmin_p \sum_{i=1}^{N} L(y_i, \hat{G}_{k-1}(X_i) + pg(X_i\mu_i))$
Output the prediction probability $\hat{G}_k = \hat{G}_{k-1} + p_k g_k(X\mu)$
End

- Extra tree
  ET is a bagging technique that uses *Gini*index to determine the best split. However, the working flow of ET is different from that of RF [29]. The differences are enlisted below.
  - In RF, the best split is used while in ET, a random split is used for information gain.
  - In RF, the bootstrapping process is used while in ET the whole original sample is used.

2) META CLASSIFIER

The ridge classifier is used as a meta-classifier in our proposed SEM [30], [31]. The three base classifiers ($B1-B3$) are trained on the dataset x_{mn} to perform the prediction after pre-processing. Each base class Bi creates new feature vectors $x'_{mn} = (B1(x_{mn}))$, $(B2(x_{mn}))$ and $(B3(x_{mn}))$ derived from the original x_{mn} vectors. After training the base classifiers, a meta-classifier is utilized to classify the newly created feature vectors. Algorithm 6 presents the proposed model's working.

V. PROBLEM FORMULATION

The classification problem is solved using the proposed model. In the first step, a matrix is used.

$$X = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1n} \\ x_{21} & x_{22} & \ldots & x_{2n} \\ & & & \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{m1} & x_{m2} & \ldots & x_{mn} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \tag{15}$$

where X stands for the entire dataset, m for observations and n for features. There exist 1034 features and 42372 observations in the dataset. The label class column represents 0 and 1 values. After selecting the features, the basic classifier ($B1 - B3$) is trained on x_{mn}. In the validation step, new features x'_m are generated from the existing ones. During this process, loss is calculated by the binary *Giniindex* [34] using Equation 16. The *Giniindex* is a two-step data distribution technique. The steps are enlisted below.

- Randomly pick the data samples from the dataset.
- Randomly perform the classification of the class distribution using Equation 16.

$$G = \sum_{i=1}^{C} p(i) * (i - p(i)) \tag{16}$$

C and $p(i)$ represent the total number of classes and the probability of the features being classified for a particular class i, respectively. The value of the *Giniindex* lies between 0 and 1.

Algorithm 6 Working of the Proposed Model

1: **Initialization**
2: The D_s dataset $x_{i,j} \in$ SGCC, where I stands for observations and j for features.
Data splitting into Training and Testing
3: The training is described by $D_t r$ while testing by x'_m
Pre-Processing
4: Normalization is performed by the min-max
5: NaN values are handled by the simple imputer
6: Data imbalance problem is solved by ADASYN-TomekLink
Base Classifiers
7: The first model is trained in the classification phase. I. **Training Phase**
8: For i=1 to S (Where S represents the whole set) on the dataset $D_t r$
9: Select S' from S using Bootstrap.
10: Model is trained on S'
11: Loss is calculated on the basis of Giniindex
12: Base Model 1 = Prediction
13: Model 2 is trained on S'
14: Model weights have been modified (Given preference to those instances which are misclassified in the previous model)
15: Loss is calculated on the basis of Ginindex
16: Base Model 2 = Prediction
17: Base Model 3 = Prediction
Meta classifier
18: The well tuned ridge classifier is tested on x'_m for final classification
19: Performance comparison is measured on the basis of valid and reliable performance metrics
20: **End**

VI. MODEL EVALUATION

The proposed model's performance is validated using various performance metrics. The validation results are discussed in detail below.

A. PERFORMANCE METRICS

Below is the discussion of the performance metrics' functioning mechanisms used in this study for model evaluation.

The AUC score is one of the most accurate and dependable performance indicators. It has a range of 0 to 1. In terms of TPR and FPR, the intra class separability is measured using this metric. The AUC's mathematical form is given in Equation 17 [8].

$$AUC = \frac{\sum Rank_{i \in positiveclass} - \frac{P(1+P)}{2}}{P * N}. \tag{17}$$

The positive class and negative class is denoted by P and N, respectively. The AUC value of 1 indicates that the model performs accurately. Whereas, random guessing is represented by the value of 0.5.

Precision and recall are used to calculate the F1-Score. It estimates the precision-to-recall ratio and measures harmonic means. For calculating F1-Score, Equation 18 is used [8].

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{18}$$

Recall and precision are calculated using Equations 19 and 20.

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$Precision = \frac{TP}{TP + FP} \tag{20}$$

The number of TP in the actual result is used to compute recall, which is also known as sensitivity. Precision, on the other hand, is also known as specificity. It is determined by the relevance of the entire actual result.

Accuracy is one of the reliable performance metrics widely used in ETD [18]. It is defined as the total number of accurately classified instances divided by the total number of instances, as given in Equation 21.

$$Accuracy = \frac{TP + TN}{TP + FP + FP + FN} \tag{21}$$

TP, FP, TN and FN are defined as follows.

TP: the users are theft and the classifier considers them as theft.

FP: the users are honest but the classifier considers them as theft.

TN: the users are honest and classifier considers them as honest.

FN: the users are theft but the classifier considers them as honest.

TABLE 2. Proposed SEM's performance comparison.

Classifier	Accuracy	ROC-AUC	Precision	Recall or DR	F1-Score	PR-AUC	FPR
RF	0.95017	0.95047	0.93289	0.96797	0.95010	0.95841	0.06693
ET	0.96109	0.96135	0.94631	0.97645	0.96115	0.96708	0.05384
XGBoost	0.86962	0.87060	0.81342	0.92097	0.86386	0.89551	0.17224
Ridge	0.72901	0.72664	0.86577	0.68471	0.76467	0.79951	0.19120
Proposed SEM	0.96382	0.96385	0.96242	0.96631	0.96436	0.97181	0.03873

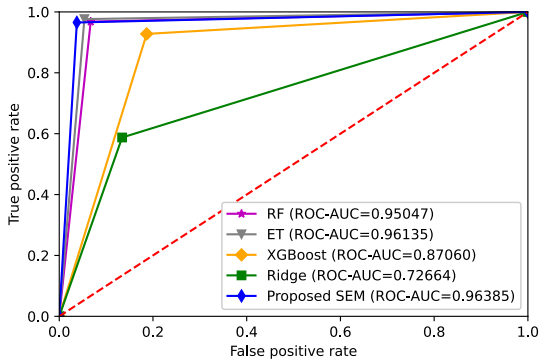


FIGURE 7. Comparison of the proposed SEM with standalone classifiers in terms of ROC-AUC.

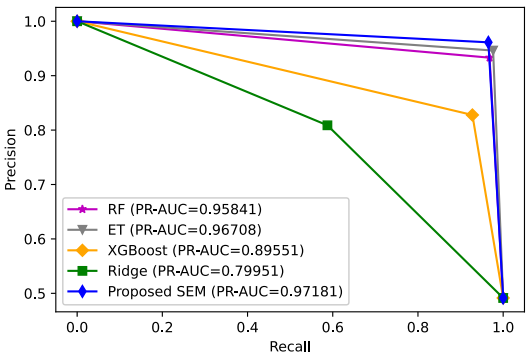


FIGURE 8. Comparison of the proposed SEM with standalone classifiers in terms of PR-AUC.

VII. SIMULATIONS AND RESULTS

This section discusses the proposed model's simulation setup and performance results obtained using various performance metrics.

A. SIMULATIONS' SETUP

The system having Intel Core i5 along with 8 Gigabytes RAM and 500 Gigabytes HDD is used for performing simulations. The proposed model is implemented using the Python programming language and well-known machine learning scikit-learn and XGBoost libraries. The Python programming language uses Google Colab as a simulation tool. By default, free-tier Google Colab provides 12 GB RAM, which is not enough to process the complete SGCC dataset comprising 42373 records. Therefore, we selected only 4000 samples out of 42372 for analysis purpose where the original classes' distribution is maintained. Afterwards, using ADASYN-TomekLink, the records are increased from 4000 to 7325 and the data is balanced.

B. PERFORMANCE RESULTS

The proposed model's performance is validated on the real SGCC dataset [21]. Table 1 provides the detailed description of the dataset. Moreover, 80% of the total dataset is used for training while 20% is used for testing. The simulation results' details are provided below.

Figure 7 shows the performance of RF in terms of ROC-AUC. The figure exhibits RF to underperform ET. It is due to the utilization of replica replacement and the splitting performed on the basis of the best split in which the whole sample is not covered. However, RF exhibits high ROC-AUC than XGBoost and ridge. Overall, the ROC-AUC of the proposed SEM is the highest of all classifiers.

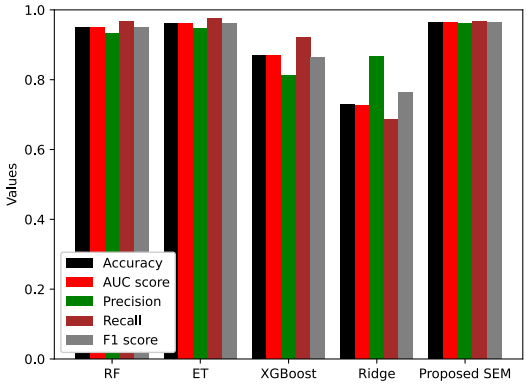


FIGURE 9. Comparison of the proposed SEM with standalone classifiers.

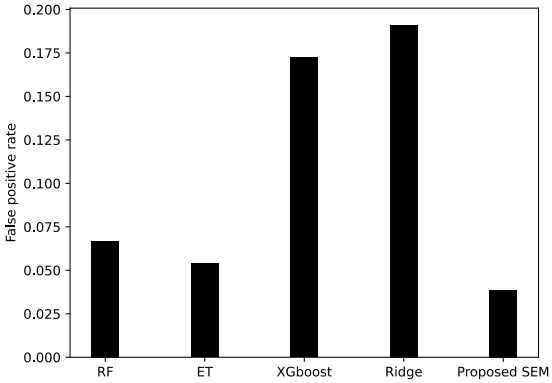


FIGURE 10. Comparison of the proposed SEM with standalone classifiers in terms of FPR.

Figure 8 presents the PR-AUC of all classifiers. If the probabilistic curve between precision and recall is high, the model is robust against the outliers and vice versa. The RF exhibits 0.01% less PR-AUC than ET. It is due to replica

replacement, which affects the model learning process and leads the classifier towards misclassification. Similarly, the PR-AUC of XGBoost and ridge is also less than that of ET. However, the proposed model's PR-AUC is the highest of all classifiers.

Figure 9 exhibits the comparison in terms of different well-renowned performance metrics. The figure shows the proposed SEM's superiority concerning the mentioned performance metrics.

The proposed SEM model's FPR comparison with the existing standalone models is shown in Figure 10. The result shows that FPR of ridge is the highest while that of the proposed SEM is the lowest.

VIII. CONCLUSION

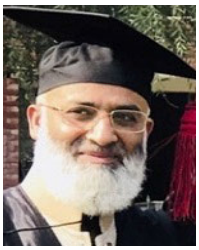
The proposed SEM consists of three classifiers at level-0 and one classifier at level-1. RF, ET and XGBoost are used at level-0 as base classifiers while ridge classifier is employed at level-1 as a meta classifier. The proposed model is three-layered architecture. The data pre-processing is done at the first layer. In the pre-processing phase, data normalization, the existence of NaN and data imbalance problems are addressed. The data normalization and NaN values are handled using min-max and simple imputer, while the data imbalance problem is handled by ADASYN and TomekLink based hybrid technique. Three machine learning models, RF, ET and XGBoost, are used at the second layer. The output of these classifiers is ensembled at the third layer to predict the final classification using ridge classifier. The dataset of SGCC is used to train and test the model. The proposed model's performance is validated using different performance metrics; the results of which reveal the superiority of the proposed model in terms of ETD and high robustness.

REFERENCES

[1] World Bank, *World Development Report 2004: Making Services Work for Poor People*, The World Bank, Washington, DC, USA, 2003.
[2] K. Sadovskaia, D. Bogdanov, S. Honkapuro, and C. Breyer, "Power transmission and distribution losses—A model based on available empirical data and future trends for all countries globally," *Int. J. Elect. Power Energy Syst.*, vol. 107, pp. 98–109, May 2019.
[3] N. Ding, H. Ma, H. Gao, Y. Ma, and G. Tan, "Real-time anomaly detection based on long short-term memory and Gaussian mixture model," *Comput. Electr. Eng.*, vol. 79, Oct. 2019, Art. no. 106458.
[4] S.-C. Yip, W.-N. Tan, C. Tan, M.-T. Gan, and K. Wong, "An anomaly detection framework for identifying energy theft and defective meters in smart grids," *Int. J. Electr. Power Energy Syst.*, vol. 101, pp. 189–203, Oct. 2018.
[5] A. Ullah, N. Javaid, A. S. Yahaya, T. Sultana, F. A. Al-Zahrani, and F. Zaman, "A hybrid deep neural network for electricity theft detection using intelligent antenna-based smart meters," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–19, Aug. 2021.
[6] S. Amin, G. A. Schwartz, A. A. Cardenas, and S. S. Sastry, "Game-theoretic models of electricity theft detection in smart utility networks: Providing new capabilities with advanced metering infrastructure," *IEEE Control Syst.*, vol. 35, no. 1, pp. 66–81, Feb. 2015.
[7] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1606–1615, Apr. 2018.
[8] W. Li, T. Logenthiran, V.-T. Phan, and W. L. Woo, "A novel smart energy theft system (SETS) for IoT-based smart home," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5531–5539, Jun. 2019.
[9] S. K. Gunturi and D. Sarkar, "Ensemble machine learning models for the detection of energy theft," *Electric Power Syst. Res.*, vol. 192, Mar. 2021, Art. no. 106904.
[10] S. Li, Y. Han, X. Yao, S. Yingchen, J. Wang, and Q. Zhao, "Electricity theft detection in power grids with deep learning and random forests," *J. Electr. Comput. Eng.*, vol. 2019, pp. 1–12, Oct. 2019.
[11] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2326–2329, Mar. 2019.
[12] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, "Electricity theft detection in smart grid systems: A CNN-LSTM based approach," *Energies*, vol. 12, no. 17, p. 3310, Aug. 2019.
[13] M. Adil, N. Javaid, U. Qasim, I. Ullah, M. Shafiq, and J.-G. Choi, "LSTM and bat-based RUSBoost approach for electricity theft detection," *Appl. Sci.*, vol. 10, no. 12, p. 4378, Jun. 2020.
[14] Z. A. Khan, M. Adil, N. Javaid, M. N. Saqib, M. Shafiq, and J.-G. Choi, "Electricity theft detection using supervised learning techniques on smart meter data," *Sustainability*, vol. 12, no. 19, p. 8023, Sep. 2020.
[15] C. C. O. Ramos, D. Rodrigues, A. N. de Souza, and J. P. Papa, "On the study of commercial losses in Brazil: A binary black hole algorithm for theft characterization," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 676–683, Mar. 2018.
[16] A. Arif, T. A. Alghamdi, Z. A. Khan, and N. Javaid, "Towards efficient energy utilization using big data analytics in smart cities for electricity theft detection," *Big Data Res.*, vol. 27, Feb. 2022, Art. no. 100285.
[17] N. Fabian, G. Figueroa, and C.-C. Chu, "NTL detection in electric distribution systems using the maximal overlap discrete wavelet-packet transform and random undersampling boosting," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 7171–7180, Nov. 2018.
[18] J. Pereira and F. Saraiva, "Convolutional neural network applied to detect electricity theft: A comparative study on unbalanced data handling techniques," *Int. J. Electr. Power Energy Syst.*, vol. 131, Oct. 2021, Art. no. 107085.
[19] R. Razavi, A. Gharipour, M. Fleury, and I. J. Akpan, "A practical feature-engineering framework for electricity theft detection in smart grids," *Appl. Energy*, vol. 238, pp. 481–494, Mar. 2019.
[20] R. Punmiya and S. Choe, "ToU pricing-based dynamic electricity theft detection in smart grid using gradient boosting classifier," *Appl. Sci.*, vol. 11, no. 1, p. 401, Jan. 2021.
[21] *State Grid Corporation of China*. Accessed: Sep. 29, 2022. [Online]. Available: http://www.sgcc.com.cn/
[22] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, 2016.
[23] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. Workshop Learn. Imbalanced Datasets*, vol. 126, Aug. 2003, pp. 1–7.
[24] B. Madhukar. (2020). *Using Near-Miss Algorithm for Imbalanced Datasets*. Accessed: Sep. 29, 2022. [Online]. Available: https://analyticsindiamag.com/using-near-miss-algorithm-for-imbalanced-datasets/
[25] J. Browniee. (2020). *How to Combine Oversampling and Undersampling for Imbalanced Classification*. Accessed: Sep. 29, 2022. [Online]. Available: https://machinelearningmastery.com/combine-oversampling-and-undersampling-for-imbalanced-classification/
[26] W. Lee. (2021). *Imputing Missing Values Using the Simpleimputer Class in Sklearn*. Accessed: Sep. 29, 2022. [Online]. Available: https://towardsdatascience.com/imputing-missing-values-using-the-simpleimputer-class-in-sklearn-99706afaff46
[27] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. Springer, 2012.
[28] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *J.-Jpn. Soc. Artif. Intell.*, vol. 14, nos. 771–780, p. 1612, 1999.
[29] F. Ceballos. (2019). *An Intuitive Explanation of Random Forest and Extra Trees Classifiers*. [Online]. Available: https://towardsdatascience.com/an-intuitive-explanation-of-random-forest-and-extra-trees-classifiers-8507ac21d54b
[30] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
[31] J. He, L. Ding, L. Jiang, and L. Ma, "Kernel ridge regression classification," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 2263–2267.
[32] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?" *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, 2004.

[33] J. Brownlee. (Aug. 27, 2020). Stacking Ensemble for Deep Learning Neural Networks in Python. MachineLearningMastery.com. Accessed: Sep. 29, 2022. [Online]. Available: https://machinelearningmastery.com/stacking-ensemble-for-deep-learning-neural-networks/
[34] A. Arif, N. Javaid, A. Aldegheishem, and N. Alrajeh, "Big data analytics for identifying electricity theft using machine learning approaches in microgrids for smart communities," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 17, Sep. 2021, e6316.
[35] A. Ullah, N. Javaid, M. Asif, M. U. Javed, and A. S. Yahaya, "AlexNet, AdaBoost and artificial bee colony based hybrid model for electricity theft detection in smart grids," *IEEE Access*, vol. 10, pp. 18681–18694, 2022.
[36] F. Shehzad, N. Javaid, A. Almogren, A. Ahmed, S. M. Gulfam, and A. Radwan, "A robust hybrid deep learning model for detection of non-technical losses to secure smart grids," *IEEE Access*, vol. 9, pp. 128663–128678, 2021.

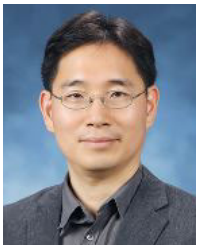
ASHRAF ULLAH is currently pursuing the Ph.D. degree in computer science with the Communications Over Sensors (ComSens) Research Laboratory, COMSATS University Islamabad, Islamabad Campus, under the supervision of Prof. Nadeem Javaid. His research interests include cloud computing, smart grids, electricity load forecasting, and electricity theft detection.



NADEEM JAVAID (Senior Member, IEEE) received the bachelor's degree in computer science from Gomal University, Dera Ismail Khan, Pakistan, in 1995, the master's degree in electronics from Quaid-i-Azam University, Islamabad, Pakistan, in 1999, and the Ph.D. degree from the University of Paris-Est, France, in 2010. He is currently a Tenured Professor and the Founding Director of the Communications Over Sensors (ComSens) Research Laboratory, Department of Computer Science, COMSATS University Islamabad, Islamabad Campus. He has supervised 158 master's and 30 Ph.D. theses. He has authored over 900 articles in technical journals and international conferences. His research interests include energy optimization in smart/microgrids and in wireless sensor networks using data analytics, and blockchain. He was a recipient of the Best University Teacher Award (BUTA 2016) from the Higher Education Commission (HEC) of Pakistan, in 2016, and the Research Productivity Award (RPA 2017) from the Pakistan Council for Science and Technology (PCST), in 2017. He is an Associate Editor of IEEE Access and an Editor of *Sustainable Cities and Society*.

MUHAMMAD UMAR JAVED (Graduate Student Member, IEEE) received the bachelor's and master's degrees in electrical engineering from Government College University Lahore, Lahore, Pakistan, in 2014 and 2018, respectively. He is currently pursuing the Ph.D. degree with the Communications Over Sensors (ComSens) Research Laboratory, Department of Computer Science, COMSATS University Islamabad, Islamabad, under the supervision of Prof. Nadeem Javaid. He has authored more than 20 research publications in international journals and conferences. His research interests include smart grid, electric vehicles, and blockchain.

PAMIR (Graduate Student Member, IEEE) received the B.S. degree in software engineering from the National University of Modern Languages (NUML), Islamabad, Pakistan, in 2016, and the M.S. degree in software engineering from the Communication Over Sensors (ComSens) Research Laboratory, Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad, Pakistan, under the supervision of Dr. Nadeem Javaid, in 2018, where he is currently pursuing the Ph.D. degree in computer science. He has authored two journals and 12 conference proceedings in international journals and conferences. His research interests include data science, smart grids, and optimal power flow.



BYUNG-SEO KIM (Senior Member, IEEE) received the B.S. degree in electrical engineering from Inha University, Incheon, South Korea, in 1998, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Florida, in 2001 and 2004, respectively. His Ph.D. study was supervised by Dr. Yuguang Fang. From 1997 to 1999, he worked as a Computer Integrated Manufacturing (CIM) Engineer at the Advanced Technology Research and Development (ATR&D), Motorola Korea Ltd., Paju, South Korea. From January 2005 to August 2007, he worked as a Senior Software Engineer at Networks and Enterprises, Motorola Inc., Schaumburg, IL, USA. From 2012 to 2014, he was the Chairman of the Department of Software and Communications Engineering, Hongik University, South Korea, where he is currently a Professor. His works appears in around 220 publications and 22 patents. His research focuses in Motorola Inc., were designing protocol and network architecture of wireless broadband mission critical communications. His research interests include the design and development of efficient wireless/wired networks, including link-adaptable/cross-layer-based protocols, multi-protocol structures, wireless CCNs/NDNs, mobile edge computing, physical layer design for broadband PLC, and resource allocation algorithms for wireless networks. He served as the General Chair for 3rd IWWCN 2017 and the TPC Member for the IEEE VTC 2014-Spring, the EAI FUTURE 2016, and the ICGHIC 2016–2020 Conferences. He served as a Guest Editor for Special Issues of the *International Journal of Distributed Sensor Networks* (SAGE), IEEE Access, and the *Journal of the Institute of Electronics and Information Engineers*. He is an Associate Editor of IEEE Access.

SAEED ALI BAHAJ received the Ph.D. degree from Pune University, India, in 2006. He is currently an Assistant Professor with the MIS Department, COBA, Prince Sattam bin Abdulaziz University, Al-Kharj, Saudi Arabia, and also an Associate Professor with Hadhramout University, Yemen. His main research interests include artificial intelligence, information management, forecasting, information engineering, big data, and information security.