Energy Reports 12 (2024) 1235–1253

Contents lists available at ScienceDirect

Energy Reports

journal homepage: www.elsevier.com/locate/egyr

Research paper

Stacked machine learning models for non-technical loss detection in smart grid: A comparative analysis

Muhammad Hashim [a,b], Laiq Khan [a], Nadeem Javaid [c,d,*], Zahid Ullah [e,*], Aymin Javed [c]

[a] Department of Electrical and Computer Engineering, COMSATS University Islamabad (CUI), Islamabad 44000, Pakistan
[b] University School of Advanced Studies, IUSS Pavia, 27100, Italy
[c] Department of Computer Science, COMSATS University Islamabad (CUI), Islamabad 44000, Pakistan
[d] International Graduate School of Artificial Intelligence, National Yunlin University of Science and Technology, Douliou, Yunlin 64002, Taiwan
[e] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milano, Italy

ARTICLE INFO

Keywords:
Smart grid
Cyber-attacks
Blackout
Advance metering infrastructure
Machine learning

ABSTRACT

The growing prominence and emphasis of renewable energy to decrease carbonization in the power system and reduce the dependability of fossil fuel for energy needs play an important role in the development of smart grids. Many technological advancements are integrated into smart grid to optimize the power system and renewable energy sources. Smart grid leverages electricity and energy consumption data exchange to establish a significantly advanced, automated, and decentralized electricity network. However, this brings numerous vulnerabilities to the power system, including cyber-attacks, grid blackouts, and electricity theft. While the most significant concern is energy theft, where some culprit's consumers manipulate their energy meters to reduce their readings. This destabilizes the country's electricity utility and economic development and causes a high tariff on energy for consumers who pay the bill. Therefore, developing an advanced framework for electricity theft detection is necessary. To address this problem, we propose a machine learning-based stacked framework to detect malicious activity in the smart grid. The proposed data-based stacked ensemble model detects honest and anomalous consumers in two stages. In the first stage, the model employs four individual classifiers at the base level to analyze data and a single classifier at the meta-level to classify the results of the base learners for the second stage classification. Furthermore, the Borderline SMOTE and Principle Component Analysis techniques are employed to address the class imbalance and curse of dimensionality issues respectively. Through experimental analysis, we proved the effectiveness of the proposed framework in detecting suspicious activity in four different experiments, including preprocessed data, feature extracted data, balanced data, and lastly, both feature engineering and data balancing. The simulation outcomes demonstrate that our proposed framework enhanced energy security and overcomes the impact of theft attacks on the smart grid.

1. Introduction

Carbonization in power systems increases the prominence and emphasis of renewable energy sources (RES). The integration of RES in power systems by leveraging smart grid helps in the optimization of energy. Smart grid integrates many modern technologies such as big data and Artificial Intelligence (AI) to optimize the RES. From another aspect, on account of energy importance and less availability of resources, the secure and efficient distribution of electricity is a crucial aspect of social and economic development in every country. Smart grid has the potential to provide a secure alternative of energy distribution and monitoring, surpassing the limitations of conventional grid systems. It integrates various sensors and computers that monitor energy distribution, consumption, control and manage consumer usage. This enables bidirectional power and information flow in the smart grid (Palahalli et al., 2019). It optimizes both the renewable energy source (solar or wind parks) and energy utilization (smart homes, smart cities, industries and charging stations) as illustrated by Fig. 1. The advancements in electrical grid systems have enabled both energy companies and consumers to monitor their energy consumption in real-time. The sensors transfer electricity consumption readings to energy utility and bill the consumers. The system's primary aim is to minimize energy losses and provide a reliable and pragmatic electricity supply.

* Corresponding authors.
E-mail addresses: muhammad.hashim@iusspavia.it (M. Hashim), laiqkhan@comsats.edu.pk (L. Khan), nadeemjavaidqau@gmail.com (N. Javaid), zahid.ullah@polimi.it (Z. Ullah), ayminjaved546@gmail.com (A. Javed).

https://doi.org/10.1016/j.egyr.2024.06.015
Received 8 April 2024; Received in revised form 26 May 2024; Accepted 5 June 2024
Available online 23 July 2024
2352-4847/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

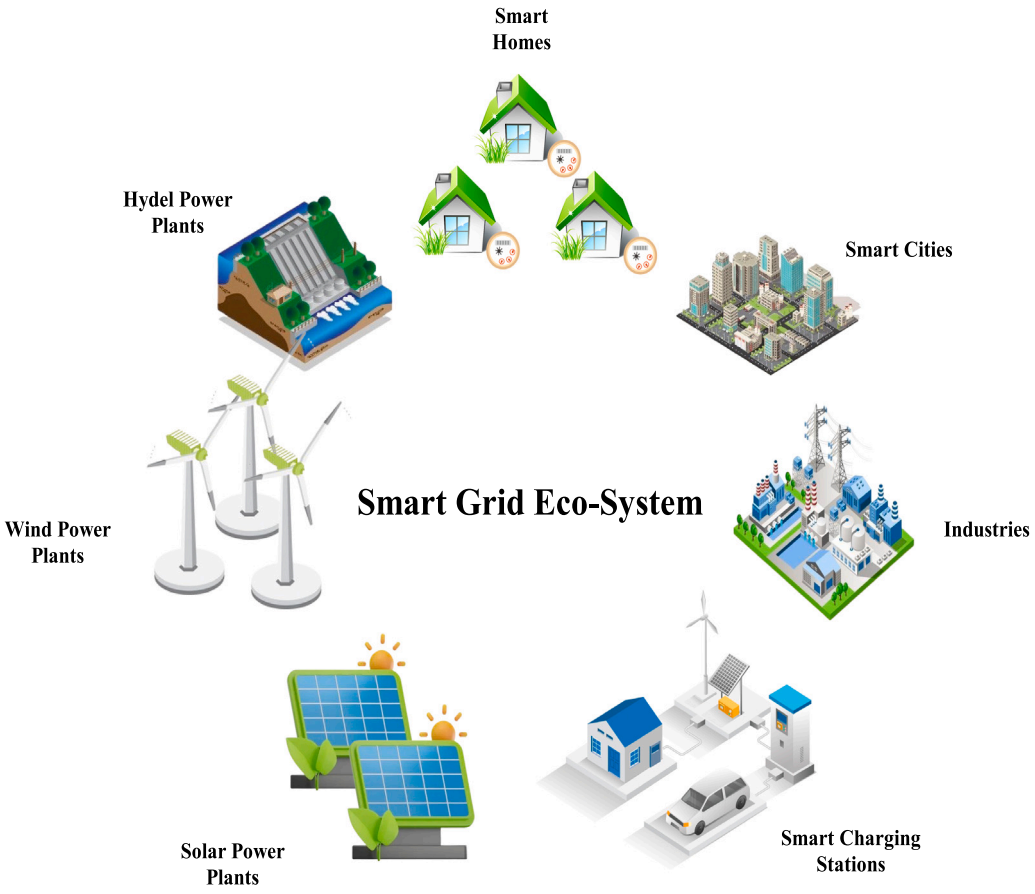


Fig. 1. Illustration of Smart Grid Eco-System.

Advanced metering Infrastructure (AMI) is an advanced version of a conventional disk type meter that measures and reports electricity consumption. It plays a very crucial role in the measurement of electricity usage and loss in the smart grid. AMI offers many advantages to electricity utilities, including real-time monitoring of consumers through electricity and information flow. It is the main component of the smart grid which gathers a huge amount of electricity consumption data, which is further utilized by electricity utility for load forecasting (Al-Turjman and Abujubbeh, 2019; Zhang et al., 2021) and demand response (Bizzozero et al., 2016). Nevertheless, due to its electronic and networking capabilities, AMI is susceptible to cyber-attacks, which can lead to blackouts, grid failures, and energy theft (Shukla et al., 2023; Shehzad et al., 2023).

The energy theft is the most severe concern for both energy users and electricity utilities. Energy theft refers to Non-Technical Losses (NTL) which include meter bypassing, false meter reading injection and network invasion (Yan and Wen, 2021). The main reason for energy theft is illegal consumers. They manipulate energy meters to lower their meter readings, with the primary objective of paying less for electricity bills. NTL jeopardizes the overall long-term viability and stability of the SG. It increases the per-unit cost of electricity for domestic and industrial consumers, which results in high production costs.

A recent study (Northeast Group, L.L.C., 2018) indicated that the world faces $96 billion of loss due to electricity theft annually as of 2018. Additionally, this is not just a problem in impoverished nations; relatively significant income losses brought on by energy theft also happen in developed nations. For instance, the revenue losses from electricity theft in the United States and China reach 6 billion dollars and 20 billion CNY respectively (Nes, 2020), Lin et al. (2021). Moreover, the steady operation of SG and consumers are both severely threatened by NTL consumers' experiences of periodic voltage dips and occasional power disruptions in locations where energy theft is prevalent, particularly during peak load hours.

These issues can lead to fires and endanger public safety in extreme circumstances. So, it is necessary to take effective measures to detect the behavior of anomalous consumers in order to protect the SG and energy utility.

The main contributions of our research are summarized below.

1.1. Contributions

- The computation overhead is a major concern when working on large datasets. So, we integrated a statistical approach called Principle Component Analysis (PCA) to extract latent features from the dataset. This reduces the computation overhead by reducing redundant information and aiding in improving efficiency and interpretability.
- In electricity consumption data, class distribution issue reduces the model efficiency and causes it to be biased towards the dominant class. We utilized Borderline SMOTE to counter the class distribution issue.
- The selection of optimal combination of the classifiers for first and second order in a stacked structure is a challenging issue. Our comprehensive experiments and results led us to the best combination of stacked models that provides the outstanding theft classification results.
- The experiment is performed in four distinct case studies to observe the behavior of the proposed system; First, the structure is assessed without Data Balancing (DB) and Feature Engineering (FE) to deeply analyze the behavior of structure only with preprocessed data.

- Second, only FE is performed on the dataset using PCA to analyze the experimental values on an imbalanced dataset.
- Third, we perform solely DB with BorderlineSMOTE on the dataset to investigate the behavior of introduced structure on balanced data and the breakdown of computation overhead.
- Finally, both FE and DB are leveraged on the dataset to get the proposed structure's experimental findings to deeply assess both methodologies' behavior.

The introduced model aims to contribute to the current research by providing an efficient framework for suspicious activity identification in SG. Through its demonstration of the efficiency of a stacked model in enhancing the precision of energy theft, the study also aims to contribute to the development of data-oriented approaches for SG systems. The rest of this research is structured as follow: Section 2 consists of related work, and the problem statement is disclosed in Section 3. Further, Sections 4 and 5 explain the proposed system model and simulation results. Finally, a conclusion is presented in Section 6.

2. Related work evaluation

NTL in power systems compromises the stability of the smart grid. The author proposed a model in Lee et al. (2022), data analysis-based electricity theft detection is one of the best solutions to reduce the issues of NTL. The fundamental issue with data-based NTL detection is that the collected energy usage data set is imbalanced. Deep reinforcement learning is applied to solve the data imbalance problem of NTL. However, compared to the conventional NTL algorithms, there is no need for extra pre-processing steps to balance the data set. The evaluation of the proposed system model is done using the True Positive Rate (TPR), False Positive Rate (FPR), False Omission Rate (FOR), and F1-score.

In Ramos et al. (2016), proposed Binary Black Hole Algorithm (BBHA) to combat NTL in Brazil. The study achieved prominent results as compared to Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). However, there is a lack of reliable performance metrics, as it is crucial while working in binary classification on imbalanced datasets. The model is evaluated on the Brazilian electricity utility's industrial and commercial datasets.

In this study (Rajiv and Choe, 2019) authors performed a comparative analysis of three updated Gradient Boosting Classifiers (GBCs), Extreme Gradient Boosting (XGB), Categorical Boosting (Catboost), and Light Gradient Boosting Machine (LGBM). Furthermore, the authors integrated six theft cases to conduct experiments with a novel feature preprocessing module. The simulation results demonstrate that the proposed method effectively detects theft from consumers.

In Aldegheishem et al. (2021), authors introduced two novel Electricity Theft Detection (ETD) methods. In the first method, a Synthetic Minority Oversampling with Edited Nearest Neighbor (SMOTEEN) and a hybrid over-sampling approach are proposed. Additionally, AlexNet is adopted to separate useful information and dimensionality reduction in energy usage data. Finally, LGBM is applied for classification of theft and normal consumers. In the second model, a Conditional Wasserstein generative adversarial network with gradient penalty is utilized to record the actual distribution of energy usage data. Additionally, GoogLe-Net architecture is adopted to minimize high dimensionality of energy consumption data. Afterwards, Adaptive Boost (ADB) is implemented for classification of theft and honest users. The experiment was carried out for both models on actual electricity usage data provided by State Grid Corporation of China (SGCC). Finally, findings demonstrate the superiority of the introduced approach in detecting theft consumers effectively, as compared to established models such as XGB, Support Vector Machine (SVM), and Convolutional Neural Network (CNN).

The author in Buzau et al. (2020), present a hybrid solution which self-learns the features and for NTL detection in SG. This hybrid method is based on Long Short-Term Memory Network (LSTM) and Multi-Layer Perceptron (MLP). In this strategy, MLP incorporates non-sequential data, like geographical location or contractual electricity, while LSTM analyze the electricity consumption. However, the proposed model has achieved 54.5% PR-AUC due to class imbalance issue in the dataset (see Table 1).

The research in Huang and Xu (2021), introduced a Stacked Sparse Denoising Autoencoder (SSDAE) based approach for NTL. Technically, the auto encoder uses the power consumption data of honest consumers as training samples, learns useful features, and rebuilds the inputs. The detector captures the theft consumers by comparing the input features of honest consumers with malicious consumers. Additionally, this method employed PSO to optimize the hyperparameters of SSDAE to enhance the efficiency and robustness of the model further.

A Deep Learning (DL) based model is used to solve the curse of dimensionality issue in Li et al. (2019). This study developed a hybrid random forest and CNN (RF-CNN) based model, where CNN captures latent features from EC data. Further, the back propagation method and dropout layer are used to update the parameters at the training stage and overcome the effect of overfitting respectively. Finally, RF classifier is used for the classification of theft and normal consumers.

Tehrani et al. (2022), presents a ML based hybrid approach to control real time large amount of SG data to combat energy theft. The research addressed the issue of imbalanced data classes by introducing an extra theft attack in addition to the six previously recognized patterns, leading to more precise classifiers. This framework is designed to boost the FPR and accuracy. However, it only achieves 88% accuracy and disregards the FPR, although it is not enhanced as typically achieved in many research works.

Research in Nazmul Hasan et al. (2019), presented a hybrid CNN-LSTM based approach. The approach is based on wide and deep CNN model to solve the one dimensional and periodicity usage problem. The wide component of CNN applied to convert the 1-D daily electricity to 2-D weekly electricity consumption data. Moreover, deep component of CNN used to solve the periodicity of normal and non-periodicity of abnormal users, which is based on 2-D energy consumption. Finally, LSTM network is applied for classification of theft and normal consumers. However, this research employed SMOTE for class imbalance issue, which causes overfitting of classifier due to similar synthetic samples generated by SMOTE.

Authors in Jokar et al. (2015), introduced a Consumption Pattern Based Energy Theft Detector (CPBETD) which leverages the normal and theft consumers profile to train the model. The proposed methodology is verified by integrating real electricity usage data of 5000 consumers and achieved good performance. NTL is major concern for power industries it compromises overall stability of the system. A Decision Tree (DT) and SVM based detector was implemented to detect malicious samples. Simulation results depicts that more than 80% of malevolent samples was correctly identified by the system (Jindal et al., 2016).

In Qu et al. (2021), proposed an ensemble DL network based on Adaboost (ADB) to counter electricity theft. Moreover, SOMTE and PCA employed to counter class imbalance and curse of dimensionality issue. The presented method outperforms the traditional classifiers such as Artificial Neural Network (ANN), RF and SVM in term of AUC on SGCC dataset. Another investigation suggested a two-step electricity theft detection system to predict the Potentially stolen electricity (PSE) to increase economic benefits in Cui et al. (2021). In the first part, a Convolutional Autoencoder (CAE) utilized to detect PSE and extract the features behavior of abnormal consumers. In second part, a transfer Xgboost (Tr-Xgboost) and transfer adaptive boosting (Tr-Adaboost) employed to learn the correlation between PSE of each consumer and extracted features by the first part.

The research in Zidi et al. (2023) introduced a dataset (ETD2022) for binary classification of theft and honest samples. This method examined sixteen types of users by using different ML techniques (KNN, DT, RF, Bagging ensemble, ANN). In Shi et al. (2023), the global features of consumption data were calculated by a Transformer Neural

Table 1
Related work.

Existing problem	Proposed methodology	Validations performed	Limitation
Class distribution.	DRL (Lee et al., 2022).	Accuracy, Precision, F1-score , PR-AUC	Less generalization of DRL.
Low detection rate, Manual inspection	Black hole algorithm, PSO, HS (Ramos et al., 2016)	Accuracy, convergence rate, execution time	Local minima, No preprocessing applied on data, Class distribution problem
High dimensionality	GBDT, weighted feature-importance (Rajiv and Choe, 2019)	Detection rate, false positive rate	insufficient performance metrics
Curse of Dimensionality	LSTM-MLP (Buzau et al., 2020)	AUC-ROC, precision recall curve	High FPR and execution time
Computation overhead, detection rate	SSDAE and PSO (Huang and Xu, 2021)	Detection rate and FPR	Low generalization ability, less performance metrics used
Low generalization	CNN-RF (Li et al., 2019)	AUC-ROC curve, precision, recall and f1-score	RF is prone to over-fitting
High FPR	Gradient boosting classifier (Tehrani et al., 2022)	Accuracy, AUC, PR Curve	Low accuracy
Curse of dimensionality	CNN-LSTM (Nazmul Hasan et al., 2019)	Precision, F1-score	Smote causes overfitting
Low DR and high FPR	Multi class SVM and one class SVM (Jokar et al., 2015)	Detection rate, FPR	Low generalization on noisy data
Low detection rate	SVM-Decision Tree (Jindal et al., 2016)	Accuracy, FPR	Low generalization on sudden changes
Low AUC score	Ada-boost and deep neural network (Qu et al., 2021)	AUC, Accuracy, sensitivity	Smote causes over-fitting
Low Accuracy	incremental Optimum-Path Forest classifier (Iwashita et al., 2021)	Accuracy	insufficient performance metrics used
High computation	Tr-AdaBoost and Tr-XGBoost (Cui et al., 2021)	Accuracy, Global error	Insufficient evaluation metrics, ada-boost prone to over-fitting
Low generalization and DR	RF-CNN-KNN (Zidi et al., 2023)	Accuracy, AUC-ROC, F-Measure	RF is too slow and Ineffective on real world data
Low TPR and High FPR	Transfer Network (Shi et al., 2023)	TPR-FPR	Insufficient data for training and lack of performance metrics
Capture 1-D data	CNN-XGB (Nawaz et al., 2023)	Accuracy, Precision, Recall, F-Score	Low AUC score, High computation
Curse of dimensionality	Ensemble Methods (Gunturi and Sarkar, 2021)	Precision, Recall, AUC	Smote causes over-fitting

Network (TNN) which further utilized for classification of suspicious and legitimate samples. The experimental results depict that proposed method provides high TPR and low FPR. The simulation results conducted on Irish dataset. The study presented in Nawaz et al. (2023), used CNN to extract the meaning full information from the data. Where the one dimensional and two dimensional electricity consumption data were fed to the CNN. Further, the extracted information was passed to the XGB ensemble model for classification. However, the XGB prone to overfitting.

The study in Almazroi and Ayub (2021) proposed a shallow network and SMOTE technique to solve low detection rate issue and class imbalance respectively. However, the integration of SMOTE causes overfitting of classifier. Further tabular literature review can be observed in table Table 1.

3. Problem statement

In literature review, different models are introduced for ETD in SG. After understanding the models, some of the problems are identified and need to be fixed. Energy consumption data contain missing and erroneous values which reduces classifiers accuracy, it is necessary to fill the missing values. Study presented in Jokar et al. (2015), introduced a consumption pattern based energy theft detection approach to detect suspicious patterns of theft consumers to counter NTL. However, the study inadequately addressed the issue of missing values. In similar study (Jindal et al., 2016), introduced SVM with decision tree algorithm for NTL detection. However, missing values are not filled properly.

Energy consumption data is commonly utilized in data based approaches for electricity theft detection in smart grid. Smart meters in AMI are employed to collect such type of data for data driven approaches. Electricity consumption data in real world cases has massive

Table 2
Data description.

Time Frame	Normal consumers	Theft consumers	Total consumers
Jan-01-2014 to Oct-31–2016	38757	3615	42372

class distribution issue, where benign samples are easily available, while theft events are rarely occurred. However, training a model on imbalance dataset, the model learns majority class samples whereas; it neglects the minority class samples. As a result, the model biased towards majority class, which causes high FPR. For the sake of unbiased and productive results, it is necessary to train the model on balanced dataset. In this study (Gunturi and Sarkar, 2021; Nazmul Hasan et al., 2019), authors utilized Synthesis Minority Oversampling Technique (SMOT) to combat unequal class distribution issue to achieve reasonable accuracy. However, SMOT algorithm randomly oversamples the theft class samples, which causes low generalization of the classifier and over-fitting.

Authors in Buzau et al. (2020), employed under sampling technique to counter the class imbalance issue. In this method some samples from majority class is discarded to balance the class distribution. However, discarding samples causes considerable information loss which reduce the overall accuracy of the proposed NTL detection architecture. Moreover, the real world electricity consumption data has high dimensionality issue means high number of features, which increases computation overhead and decreases accuracy of the classifier. In addition, the research in Gunturi and Sarkar (2021),Buzau et al. (2020) and Li et al. (2019) did not perform any feature engineering to address the curse of dimensionality in the dataset. The high dimensional data set increase computational overhead decreases the model's accuracy. In this study (Nawaz et al., 2023), a CNN model is utilized to extract the latent features from electricity consumption data. However, CNN model is designed to work with grid-like topology and works best in images data, and not suitable for time series electricity consumption data as it has sequential structure.

4. Proposed system model

We proposed an Ada-boost based ensemble stacked model to detect electricity theft in smart grids. Fig. 4 exhibits proposed system model, which comprises six modules: data acquisition, pre-processing, feature engineering, data augmentation, data splitting finally classification and evaluation of proposed system model. A detailed overview of these six modules are as follows. Fig. 2 presents the flow of proposed framework.

4.1. Data acquisition

The preliminary experiments were carried out on verified electricity usage data of 42,372 energy consumers over 1035 days. As presented in Table 2, the energy consumption data comprises 38,757 honest users labeled as "0", while the rest of 3615 users are anomalous and labeled as "1". The "FLAG" column in the dataset contains labels for electricity theft and normal consumers. The class difference depicts that there is a significant class imbalance issue which, results in unsatisfactory experimental results. So, it is necessary to resolve the class imbalance issue by employing adequate techniques. In order to investigate the distinctive patterns of energy consumption for honest and theft users, separate curves for both consumers are plotted by examining the data for 30 days from the energy consumption data (Zheng et al., 2018). Fig. 3a represents the pattern for normal user which varies between 4 to 8 kwh and exhibits a constant periodicity. Furthermore, the pattern for theft consumers shows enormous variations throughout 30 days' period in Fig. 3b. Furthermore, the dataset contains missing and erroneous readings, making pre-processing of the data is essential in order to adequately investigate the underlying hidden patterns in the dataset (see Fig. 3).

4.2. Pre-processing

The data pre-processing step is particularly important in data because the realistic EC data consists of missing values which degrade the performance of the classifier.

- **Handling Missing Values:** The data set contains some missing values which are replaced with not a number (NaN) or empty space. This happens due to smart meters failure fault in distribution line and data storage system. Data set with missing values highly affects the performance of model. The simple imputer technique is applied to remove the missing values. This study recovers missing values by integrating simple imputer method in Zheng et al. (2018). Following equation (1) represents the working of simple imputer technique.

$$F(x_{i,t}) = \begin{cases} \frac{x_{i,t-1}+x_{i,t+1}}{2}, x_{i,t} \in NaN, \\ Fx_{i,t}, \qquad else, \end{cases} \tag{1}$$

- **Data Scaling:** The primary goal of data scaling is to transform data into a specific range as ML and DL methods are sensitive to diverse dataset. Make sure that all attributes are standardized to a consistent scale, while preserving the relative differences in the values within dataset while, performing data scaling process. The following formula is used for data scaling.

$$Fx_{i,t} = \frac{x_{i,t-1} + x_{i,t+1}}{max(x_{xi,T}) - min(x_{xi,T})} \tag{2}$$

- **Outliers Treatment:** There are some erroneous values in raw data such as outliers, and these values correlate to the peak hours of energy usage. These values occurred due to high usage of energy during holidays. In this research we employed Local Outlier Factor (LOF) (Yeckle and Tang, 2018) to remove the erroneous values in the raw data. This method considers the density of the given sample and the density of the data sample in the k-nearest neighbor set. Outliers are found by contrasting each samples local density with that of its neighbors. It is significant to remember that the choice of k affects the output factor and that LOF computations work effectively.

4.3. Feature engineering using principle component analysis

The energy consumption data consists of large number of features which refers to curse of dimensionality. This increase the computation overhead and reduces the generalization ability of classifier. PCA is a widely used technique in image processing, data compression and time series analysis. In SG, PCA (Musleh et al., 2019), is used to reduce the high dimensionality of data to improve the generalization ability of classifier and reduces computation overhead. Generally, a real-life data set contain high number features, and this makes difficult for a ML algorithm to process the whole data set. It makes the classifier slow and creates over-fitting problem. However, it is very important to minimize the high dimensional data to make our classifier faster and enhance performance. Data scaling and data normalization is performed before applying the PCA on the data set. Data scaling is performed to scale the data in a specific range while data normalization is performed to change the shape of data distribution. This approach best demonstrated the variance distribution by revealing the hidden, intricate, and internal structure of the measurements set. The primary benefit of PCA is the ability to reduce measurement set dimensions while maintaining variance between measurement points. PCA can provide a lower dimensional representation of that space because each variable is associated with an axis in the higher dimensional data space. Afterwards, the value of covariance matrix, eigenvalues and eigenvectors are calculated. This step performed to select the components from original features vectors to form a new feature vector. By forming principal components with all the calculative values this process comes to an end.

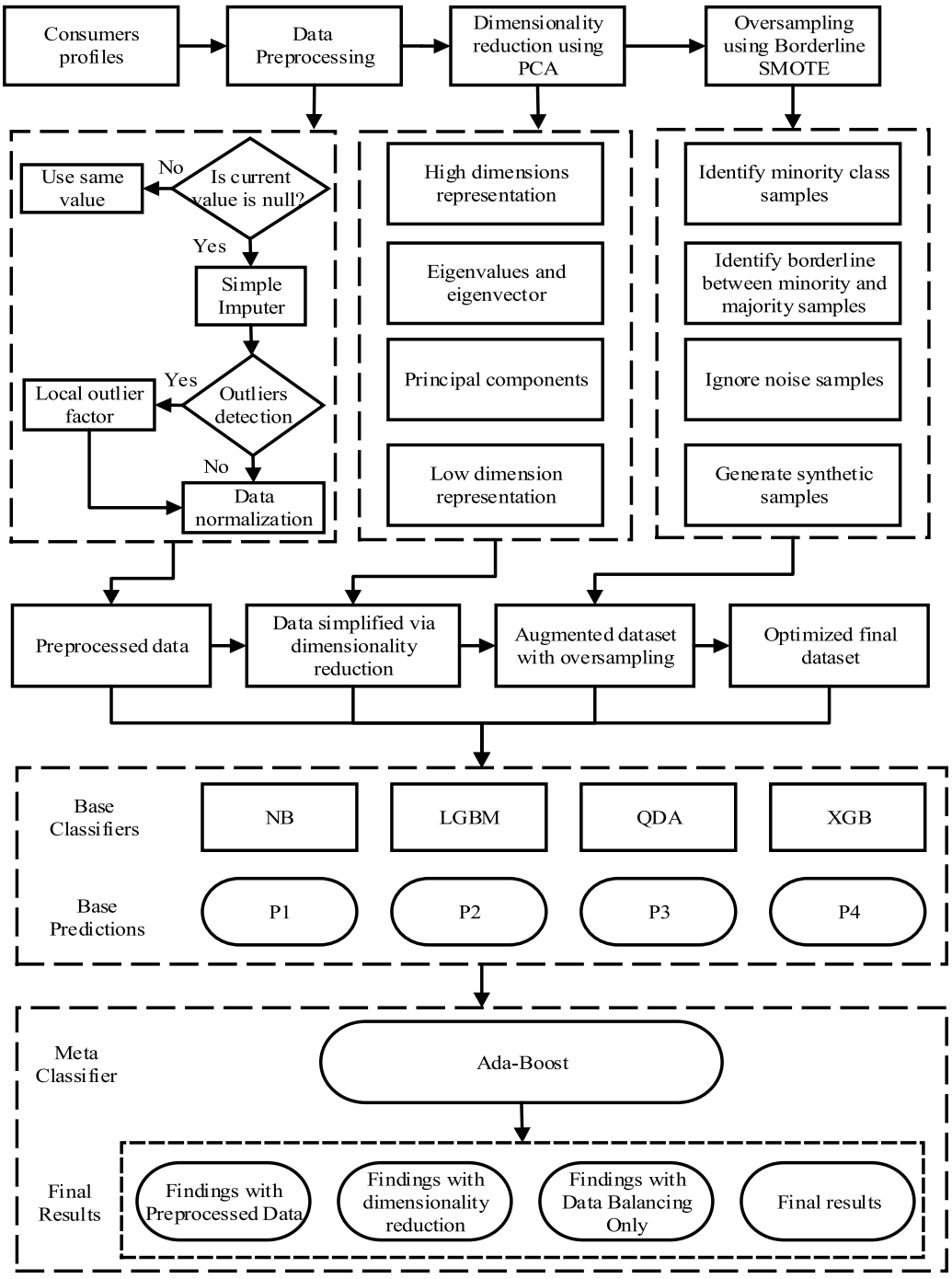


Fig. 2. Flowchart for Proposed System Model.

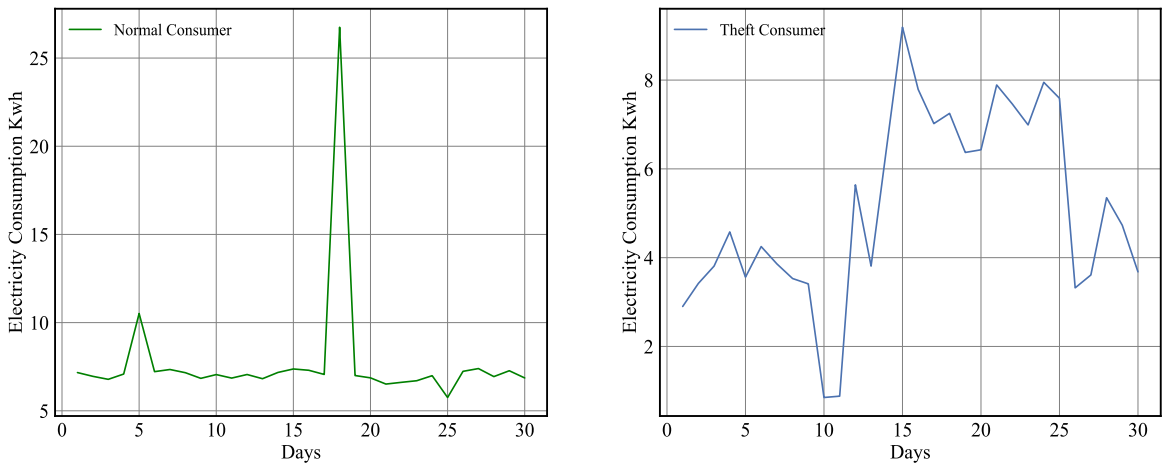
Given a data matrix X with m samples and n features, we aim to analyze the covariance structure between n features. PCA is used to generate n PCs, where each PCs is a linear combination of the n features. All PCs are mutually orthogonal and the primary objective of using PCA is that the first few PCs capture a significant portion of the variance in data matrix X enabling us to assess the relationships among the features in the data matrix X. Initially, we focus on the linear function $a_1^T X$ that depicts the highest variance between features.

$$a_1^T X = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = \sum_{j=1}^{n} = a_{1j}x_j$$

Where, vector a_1 is composed of a set of n features $(x_{11}, x_{22}, \ldots x_{1n})$, and T denotes transposition of vector a_1. Using a similar approach, we identify the second linear function X with highest variance and it is orthogonal to the first linear function X.

$$a_1^T X = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = \sum_{j=1}^{n} = a_{1j}x_j$$

Where, vector a_1 is composed of a set of n constants $(x_{11}, x_{22}, \ldots x_{1n})$, and T denotes transposition of vector a_1. As a result, the second PCs preserve the second highest variance and the coefficients w2 are orthogonal to w1. We compute n PCs in ascending order according



(a) Honest consumer (b) Theft consumer

Fig. 3. Electricity consumption of normal and theft consumer.

to highest variance. The detailed working of the PCA are shown in algorithm 1.

Algorithm 1 Principle Component Analysis

Initialize Algorithm

1: **Input:** Training data
Output: Results
2: Load data set
3: Training samples S =S_1,S_2,...,S_n ⇒ S = Number of Samples
4: covariance matrix $\sum_z$
5: **while** new samples received, **do**
6: Calculate the mean μ
7: **for** i = 1-n, **do**
8: Calculate $(z_i - \mu)(z_i - \mu)^T/n$
9: Add $(z_i - \mu)(z_i - \mu)^T/n$ **to covariance matrix** $\sum_z$
10: **end for**
11: Return covariance matrix $\sum_z$
12: Calculate eigenvectors **V** from $\sum_Z$ **V** = $V\wedge$
13: Find the score of anomalies ψ
14: **end While**
15: Return score of anomalies ψ

4.4. Addressing class imbalance

It is the key feature of smote to over-sample the minority class samples. However, SMOTE overlap samples between theft and normal class and it causes erroneous classification of normal and suspicious consumers. However, in scenario where minority data points projected entirely in the majority class, borderline smote consider these data points as noise and exclude them from oversampling process. Typically, in real world electricity data honest samples are easily accessible whereas, dishonest users are seldom encountered. However, training a classifier on imbalance data, the classifier is inclined to dominant class while ignoring the minority class samples. This leads to high False Positive Rate (FPR) and compromises the classifier's overall performance (Yao et al., 2021). The following outlines the working steps of smote borderline.

Step1: For each sample x in the positive sample set, euclidean distance is calculated between it and each other sample in the positive sample set and negative sample set, and m nearest neighbor samples are found, marked as x_i, i ϵ 1,2,3, ...,m;

Step2: Divide positive samples into different types. For positive sample x, assuming that n out of m nearest neighbor samples belong to negative samples, if n = m, sample x is considered as a noise sample; if 0 ≤ n ≤ m/2, then sample x is regarded as a safe sample; if m > n ≥ m/2, x is a boundary sample. Boundary sample set is border = $b_1, b_2, \ldots, b_{num}$;

Step3: For b_i in Border, calculate its k nearest neighbor samples in positive sample set and randomly select s samples $p_1, p_2, \ldots, p_s$ from its k nearest neighbor samples. Then, s random numbers between 0 and 1 $r_1, r_2, \ldots, r_s$ are generated for synthesizing s new samples:
bij = $b_i + r_j \cdot (b_i - p_j)$

Step4: Combine new samples into training set. Though Borderline SMOTE method does not synthesis new samples for noise, greatly reducing the probability of introducing new noise, some boundary samples may still synthesize the noise samples.

The above working steps only over-sample instances from minority class which are projected in borderline. The Borderline SMOTE employed SVM algorithm to select the best hyperplane that separate both classes with maximum margin. The optimal hyperplane is only found based on a few samples called support vectors. The Borderline SMOTE incorporates interpolation and extrapolation to over-sample minority class examples that projected near the borderline. The algorithm calculate the degree of oversampling based on the number of nearest neighbors of the majority class near the support vectors of minority class. This method creates synthetic samples using SMOTE interpolation, if the majority class makes up the majority of the m nearest neighbors of the selected minority support vector. However, the technique uses extrapolation to apply SMOTE over-sampling if fewer than half of the m nearest neighbors belong to the dominant class. Further, deep working of Borderline SMOTE can be found in Han et al. (2005).

5. Proposed methodology for NTL detection

This section of research summarizes the comprehensive overview of the proposed method for stacked generalization.

This structure is followed by NTL detection in the SG. The EC data are cleaned, normalized, balanced and finally ready to train the model. In context of model selection, an ensemble stacked model are selected, contestable it is the best model compared to the previously available models, by proving its ability by winning many classification competitions of Netflix and Kaggle recently (Džeroski and Ženko, 2004), Tang et al. (2014). Furthermore, the appropriate selection of base-classifiers for first stage and meta-classifier for second stage classification are involved.

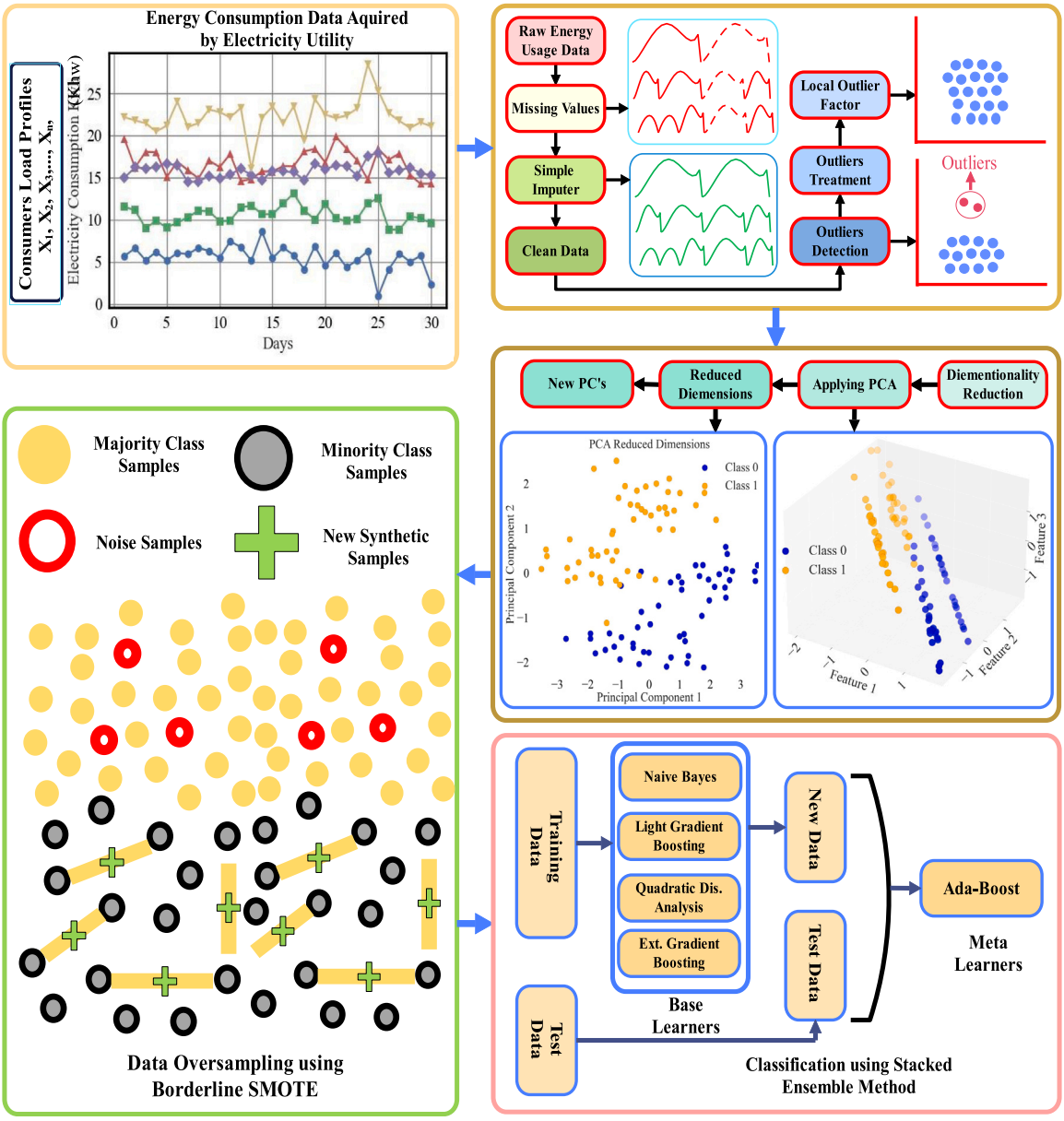


Fig. 4. Illustration of proposed system model for electricity theft detection.

5.1. Key concept of ensemble stacking

Ensemble stacking performs Machine Learning (ML) tasks by assembling and combining many learners and often referred to as two stage architecture. Essentially, a group of single learners is constructed for first stage classification at base level, and these results are then combined for second stage classification using a specific strategy. The initial step is to gather the predictions from each model at base level to form a new dataset. This dataset consists the previous first stage predictions with their actual labels for second stage classification. The newly formed data is considered as a new learning problem and a model is employed to solve this problem (Ting and Witten, 1997). The integration of stacked ensemble model for learning problems, offers frequently better generalization performance and accuracy compared to the single learner based approach. Fig. 5 represent the typical structure of ensemble stacked architecture.

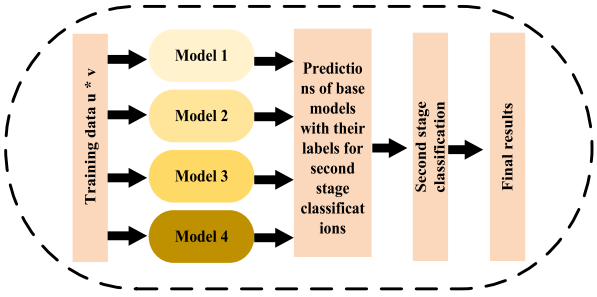


Fig. 5. Typical framework of ensemble learning.

5.2. Integrated stacking architecture

The research in Ting and Witten (1997) introduced a novel method for combining a group of single learners at base level and train a single classifier at meta level on the predictions of base level. This

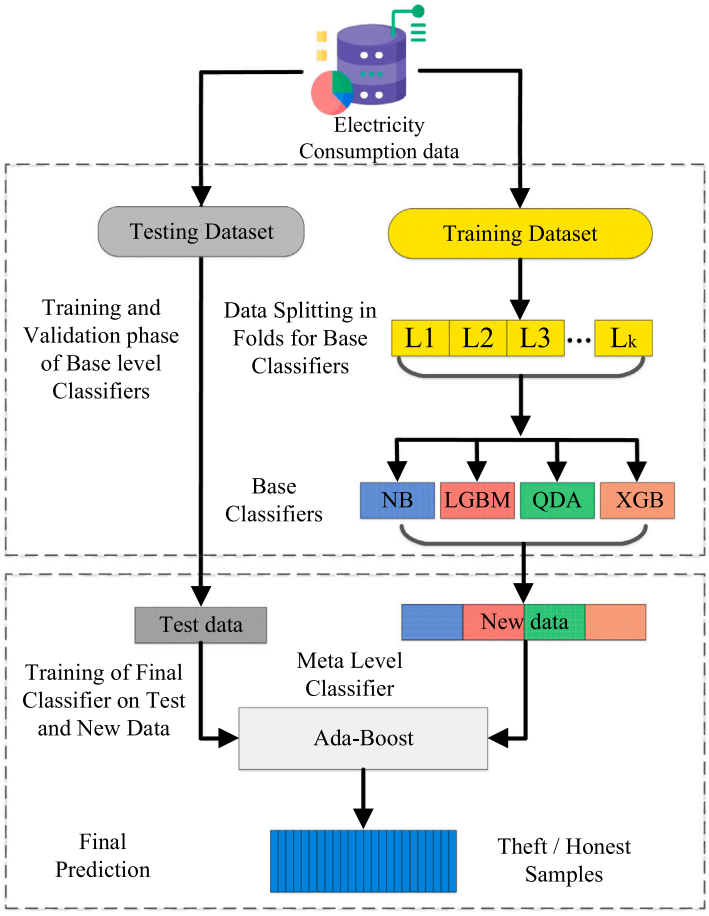


Fig. 6. Structure of stacked ensemble method.

method has three primary stages, first stage is involving with splitting the preprocessed dataset Dn = (Yn, Xn), n=1, …, N, into training and testing set, where Yn represents the class labels and Xn corresponds to the feature vector of *n*th samples. The training dataset Sn = (Yn, Xn), n = 1,2, …, N is further partitioned into L-fold cross-validation (L1, L2, …, Ln), while the test dataset T = (Xq), q=1, 2, …, Q is further used for testing the meta classifier. Fig. 6 shows the architectural working of stacked ensemble method.

In the second stage, a number of H single learning algorithms are selected for the base level (often denoted as level-0 classifiers), which are used to train the base level model of the stacked generalizer. For each base classifier H1, H2, …, MP, a separate training process k is carried out. During each iteration of the training phase for the base level classifiers, 1/k of the samples are separated for the testing phase to make predictions for the meta level classifier. Let Vk(X) be the predictions of base level classifiers (Hk) on data X, and Zkn = (Vk)(Xn) be the data acquired from the predictions of the H models after completing the cross-validation process, given by

$$Jcv = Yn, z1n, \ldots, zkn, n = 1, \ldots, N \tag{3}$$

This data is then used for meta level classification. For the meta level classification, a single model N is employed to complete the final classification. A simple classifier is used for the final classification, while strong models are used at base level to avoid over-fitting. In this framework, the Ada-boost classifier is employed for meta level classification. The algorithm Xia et al. (2022) represents the working of ensemble stacked generalization (Xia et al., 2022).

Algorithm 2 Ensemble Stacking

Input: Training dataset D = (X_1,Y_1), (X_2, Y_2), … , (X_n, Y_n); Base Level Classifier L1, L2, … , Ln; Meta Level Classifiers J.

for t = 1,2, ..., T **do**
  $h_t = L_t(D)$;
end for
D'= ϕ
for i = 1,2 ... ,m **do**
  for t = 1,2 ... ,T **do**
    $z_i t = h_t(x_i)$;
  end for
  D'= D' U $((z_i 1, z_i 2, ..., ziT), y_i)$;
end for
h'= L (D')

Output: H(x) = h' $(h_1(x), h_2(x), ..., h_T(x))$

6. Base level classifiers

6.1. Naive Bayes

Naive Bayes (NB) is a famous supervised ML algorithm (Gupta et al., 2021), widely used for classification and regression problems. NB make assumption that all features in the dataset are independent, meaning that each feature is not influenced by any other feature in the whole dataset. The algorithm's primary working depends on the target variable's posterior probability. It works on bayes theorem and

represented by the following equation.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \quad (4)$$

Algorithm 3 Naive Bayes Algorithm

Input: Training data
Output: Results

1: Initialize Algorithm;
Step 1
2: Let D be a training set of a tuple and their associated class labels. In our case, a tuple is a packet and it is represented by an n-dimensional attribute vector X = (x1, x2, x3,…, xn).
Step 2
3: The number of class m is equal to 2 as we have attack class and normal class. Given a tuple or a packet X, NB will predict that X belongs to the class with the highest posterior probability. In other words, X belongs to Ci if and only if:
$(P(C_j|X) > (P(C_j|X) for 1 \le j \le m, j \ne i$
So we maximize the $P(C_i|X)$which is given by the bayes theorem:
$P(C_j|X) = \frac{P(X|C_i).P(C_i)}{P(X)}$
Step 3
4: Because P(X) is constant for all classes, we remove it then maximize only the P(X|Ci).P(Ci). The prior probability P(Ci) is given by:
$P(C_i) = \frac{|C_i,D|}{|D|}$
5: Where $|Ci, D|$ is the number of training tuple of class C_i in D.
Step 4
6: As NB assumes that there is no relationship among child nodes or attributes, the P(X|Ci) is given by:
$P(X|C_i) = \Pi n_{K=1}\ P(X_k|C_i)$
7: Where X_k is the value of attribute A_k for tuple X. As the attributes $A_1, A_2 \ldots A_n$ in the data set are categorical, the $P(x_k|C_i)$ is the number of tuples of class C_i in D having the value X_k for A_k, divided by $|C_i, D|$.

We convert the above equation for clearer perception by replacing B with X (input attributes) and A with y (target variable) to solve the occurrence of y, given the input attributes X.

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)} \quad (5)$$

For simplifying Naïve assumption that all features are independent, regardless of which class they belong, so P(X|y) can be expressed as:

$$P(y|X) = P(x_1|y) * P(x_2)|y * \ldots * P(x_n|y) \quad (6)$$

In a probabilistic algorithm, the primary objective is to find the probability of target variable y given input variable. So, input feature P(X) is a constant and the above equation be defined as:

$$P(y|X) \propto P(y) * \Pi_{i=1}^{n} P(x_i|y) \quad (7)$$

The main objective of NB is to find the maximum probability y target class. Here, argmax function is a mathematical operation that determines the input that results in the maximum output value of target function.

The algorithm Gupta et al. (2021) delineates the detailed working of NB algorithm.

6.2. Light Gradient Boost Machine (LGBM)

LGBM algorithm is a famous boosting algorithm, presented by Guolin Ke in 2017 (Ke et al., 2017). LGBM is an improved version of gradient boosting algorithm that integrates the capability to prioritize electricity consumption samples with larger gradients and execute feature selection (Brownlee, 2021). LGB algorithm splits the tree leaf-wise rather than depth wise, which increases the prediction accuracy and improve the training speed of the classifier. Gradient One Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) techniques enables LGB more efficient compared to the other boosting technique. GOSS exclude samples having low gradients and emphasizes the samples with higher gradients in order to approximate the information gain from training instances.

Algorithm 4 Light Gradient Boosting Machine

Input: Training data
Output: Results

1: Load dataset
2: Combine samples that are mutually exclusive by EFB
3: Set θ_0(X) = $argmin_c \sum_i^N L(y_i, c)$;
for m = 1 to M **do**
4: Calculate gradient absolute values:
$r_i = |\frac{\delta L(y,\theta(x))}{\delta\theta(x_i)}|_{\theta(x)=\theta_{m-1}(x)}$, i = 1,…,N
5: Resample data by GOSS process: high N = f * len (T); randN = z * len(T); sorted = GetsortedIndices(abs(r));
6: Calculate information gain by:
$V_j(d) = \frac{1}{n}(\frac{(\sum_{Xi\epsilon Fl} r_i + \frac{1-a}{b} \sum_{Xi\epsilon Z_l} r_i)}{n_l^j(d)})^2 + (\frac{(\sum_{Xi\epsilon Fl} r_i + \frac{1-a}{b} \sum_{Xi\epsilon Z_l} r_i)}{n_l^j(d)})^2$
7: Create a novel decision tree θ_m (X)' on Set T'
8: Update θ_m(X) = θ_{m-1}(X) + θ_m (X)
9: **End For]**
10: Return θ (X) = θ_m (X)

As the samples with higher gradient plays a vital role in information gain's approximation. By prioritizing examples with higher gradients, GOSS accelerates learning process and reduces the computational overhead. EFB technique is employed for feature bundling of commonly same samples in the data set. Basically, EFB reduces the effective samples from training data and improve efficiency and reduce dimensionality. The working of LGB (Taha and Malebary, 2020) is presented in Algorithm 4.

6.3. Quadratic Discriminant Analysis (QDA)

Quadratic Discriminant Analysis (QDA) is a ML algorithm used to classify observations into their separate classes based on their input variables (Mabunga et al., 2020). QDA is a generative model that models the Probability Density Functions (PDF) of each class using a quadratic function. The PDF of each class indicates the likelihood of an observation belonging to that class, given the values of its input variables. In QDA, the class-specific prior is simply the proportion of data points that belong to each class. Following is the working steps of QDA:

- Using a quadratic function, calculate the probability density function P(X|y) for class labels y given a training dataset with input features X and class labels y.
- Calculate each class's prior probability P(y).
- Determine the posterior probability for each class p(y|x) using Bayes' theorem:

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)} \quad (8)$$

- Predict the class label y^* that increases the posterior probability of the input sample X:

$$y^* = argmax(p(y|x))\ for\ all\ y \quad (9)$$

It creates complex decision boundaries because of it non-linearity function. However, it might experience over-fitting, when the number of training examples are lower in comparison to the number of input variables. Detailed working of QDA technique can be observed in algorithm 5 (Anagnostopoulos et al., 2012).

Algorithm 5 Quadratic Discriminant Analysis

Input: Training data
Output: Results

1: Load dataset
Step 1
2: Training Data samples S and target variable V
3: Assumption of class-conditional densities of Gaussian distribution $P(x|t = c, \mu_c, \sum_c) = N(x|\mu_c, \sum_c)$ $Hereclass - specificcovariancematrix = \sum andclass - specificmeanvector = \mu$
Step 2
4: Find posterior probability using Bayes theorem,
5: P(t= $c|x, \mu_c, \sum_c$)= $\frac{P(x|t=c,\mu_c \sum_c)P(t=c)}{\sum_{k=1}^{K} P(x|t=k,\mu_k,\sum_k P(t=k)}$
6: separate class by x
h(x) = argmax $P(t = c|x, \mu_c, \sum_c)$ (3)
7: $log(P(C = c|X = x))log(P(C = c)) - \frac{1}{2}log \sum_c - \frac{1}{2}(x - \mu_c)^T \sum_c^{-1}(x - \mu_c)$
Output: Decision Boundreies

Algorithm 6 Extreme gradient boosting

Input: Training data
Output: Results

1: Load dataset
2: Initialize f_0(x);
for k = 1,2,...,M **do**
3: Calculate gk = $\frac{\delta L(y,f)}{\delta f}$;
4: Calculate hk = $\frac{\delta^2 L(y,f)}{\delta f^2}$;
5: Determine the structure by choosing splits with maximized gain
6: A = $\frac{1}{2}[\frac{G_L^2}{H_L} + \frac{G_R^2}{H_R} - \frac{G^2}{H}]$;
7: Determine the leaf weights $w^* = -\frac{G}{H}$;
8: Determine the base learner$b(x) = \sum_{j=1}^{T} wI$;
9: Add trees $f_k(x) = f_{k-1}(x) + b(x)$;
10: **End for**
11: **Output:** $f(x) = \sum_{k=0}^{M} f_k(x)$

6.4. Extreme gradient boosting

XGB is an ML based supervised learning algorithm introduced by Tianqi Chen and Carlos Guestrin Anon (2023b). It is a tree-based ensemble technique used for classification and regression. As shown by name the XGB is boosting algorithm and widely adopted by real-world applications (Hu et al., 2019). Like other boosting algorithm XGB make powerful model by creating the week learners strong. In boosting algorithms weak learners become more powerful by improving the residual of previous week learners using loss function. It overcome the residual in the predictions by creating new trees from existing weak learners in the model. Succinctly, XGB create trees sequentially to overcome the residuals in the model. It calculates the average of the target function and find prediction of the target variable. Afterwards, it calculates the residual of target feature using average of the features and construct new weak learner for the attribute and splits the tree. A detailed introduction of XGB as follows (Choi, 2019).

a. Objective function of XGB Commonly, the sum of loss function (L) (which differentiate between predicted and actual values) and regularization terms (w) (which automatically defined) over the parameters (0) are the general objective functions in ML.

$$Obj(\theta) = L(\theta) + W(\theta) \tag{10}$$

The objective function for XGB is derived from above equation which combines the sum of certain L which evaluated over all samples and the sum of regularization term for all DT.

$$Obj(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \omega(f_k) \tag{11}$$

Where y_i is the actual labels of total labels, $\hat{y}_i$ is the predicted outcomes and f_k is the number of decision tree. The primary objective of the algorithm is to reduce the objective function in the equation as low as possible. The regularization term controls the tree's complexity by varying the tree structure's depth, size, and other characteristics.

b. Base Learner

Boosting algorithms are begins with base learner. It is referred to as a "base" because the ensemble algorithms start with first model and a "learner" because all other models learn from residuals of itself. All boosting algorithms work on residual, this makes the weak learner strong simultaneously by reducing the residual.

c. Addictive model training

Typically, model optimization is performed after training but in XGB the model is trained after every iteration and this method is called addictive way of model training. The addictive way of training makes XGB more robust as compared to other ML algorithms.

d. Ideal selection of tree structure

The scoring function below is used to determine the optimal DT structure among an infinite number of possible structures.

$$Gain = \frac{1}{2}[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}] - \gamma \tag{12}$$

Where, $\frac{G_L^2}{H_L+\lambda}$ and $\frac{G_R^2}{H_R+\lambda}$ are the values of left and right leaf after splitting the node respectively; $\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}$ is the values of left and right leaf with out splitting. The tree with maximum gain is selected to select the ideal tree structure.

7. Meta level classifiers

7.1. Adaptive boosting

Adaptive boosting (Ada-Boost) algorithm is a famous boosting algorithm. It can be used for both classification and regression problems. Basically, it creates decision stumps with two stages to avoid over-fitting problems. The records are classified on the base of decision stumps. These decision stumps consist of only two nodes, the parent node, and the child node. Initially, ada-boost (Zeng et al., 2020) create decision stumps for all samples and assign equal weights by ($\frac{1}{totalnumberofsamplesN}$). Afterwards, it makes predictions and calculate total error between predicted and actual values of all stumps.

Algorithm 7 AdaBoost

Input: Training data
Output: Results

1: Load dataset
Assign initial weight to all samples (S) in dataset w_i = 1/N, i= 1,2,...N. ⇒ S = Number of Samples
2: For m = 1 to M:
3: Pass the training data G_m (x) to the classifier
4: Compute

$$Error_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i} \tag{1}$$

5: Compute [α_m = log((1- $Error_m$)/$Error_m$)]
6: Set $w_i \leftarrow w_i . exp\ (\alpha_m . I\ (y_i \neq G_m\ x_i))$, i = 1,2,...,N.
7: Output G(x) = sign($\sum_{i=1}^{N} \alpha_m\ G_m$ (x))

Then it updates the weights of misclassified samples by ($w_i = W_{i-1} \times e^{+}1$) and update the weights of the right classified samples by

($w_i = W_{i-1} \times e^{-1}$). Normalize the updated weight and create next stump. By normalizing the weight, create a new data set based on same size, this increases the likelihood of selecting the misclassified records. The detailed algorithm of ada-boost can be found in Anon (2023).

8. Performance metrics

Selecting appropriate performance metrics for evaluation of ensemble model is essential and challenging task. However, following are different performance metrics we use to evaluate our system model.

- **Accuracy** It is most important metric while Classifying between two classes. It is the ratio between true positive and true negative to the total predictions of the classifier. The value of accuracy is calculated by Eq. (13).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{13}$$

- **Precision** It is calculated by number correct positive predictions divided by total number of positive predictions (true positives and true negatives). Mathematically, it is calculated by Eq. (14).

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

- **Recall** It is the percentage of correctly classified samples by the model from class of interest (Positive class) out of total samples in the positive class and calculated by the following formula. (15).

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

- **F1-Score** F1-measure is the harmonic mean of precision and recall; it gives equal weight to the precision and recall. It means the model obtain high f1-measure if both values are high.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{16}$$

- **Area under the curve (AUC)** AUC (Bradley, 1997) is the most important metric and widely used method while dealing with imbalanced dataset. It is the likelihood of randomly selected negative sample ranks lower then a randomly selected positive sample. Mathematically, it is follows as:

$$\text{AUC} = \frac{\sum_{i \in F} \text{rank}_i - M(1 + M)/2}{M \times N} \tag{21}$$

Where, M and N are the theft and honest consumers respectively; rank_i is the rank value of sample i. F is the set of positive samples.

- **AUC-ROC Curve**
The Receiver Operator Characteristic Curve (ROC Curve) is a graph that displays how well a classification model performs across all classification thresholds. This graph shows two parameters (TPR and FPR).
- **Area under the Precision Recall Curve (PR-AUC)**
The precision (positive prediction) is calculated by dividing the total number of positive predictions by the proportion of true positive. Precision measures the percentage of accurate positive predictions. However, recall determines the ratio between positive (theft) samples classified as positive (theft) from all positive samples. PR curve shows that how good a model at classifying between two classes.
- **Confusion Matrix** It presents predicted values and actual values by N x N matrix to evaluate the performance of classifier, where N is the actual number of target classes. Following are the main components of confusion matrix as shown in Table 3.

Table 3
Confusion matrix.

Binary classes	Actual positive label (1)	Actual negative label (0)
Predicted Positive label (1)	TP	FP
Predicted Negative label (0)	FN	TN

True Positive (TN): It shows the number of positive samples correctly classified as positive. If the consumer profile is labeled as 1 and model predict as 1, then this type of outcomes considered as 'TP'.

True Negative (TN): Represents the number of negative samples identified as negative. If the model predict the outcome as 0 and the actual label of sample is 0, then it is represented as 'TN'.

False Positive (FP): The model predicts incorrectly a sample from negative class as positive sample. It represents type 1 classification error and is the opposite of 'TN'.

False Negative (FN): It represents that how many sample model incorrectly identified as negative from positive class. If the actual sample label is 1 and model predict as 0, then this type of predictions considered as 'FN'.

- **Mathew Correlation Coefficient (MCC):** MCC is used to measure the prediction quality of binary classification model. It presents the combination of all four values employed in confusion matrix and provide a balanced model evaluation while working on imbalanced datasets. The MCC ranges between −1 to 1, where 1 denotes a perfect model prediction, 0 represents random prediction and −1 shows the utter discrepancy between predictions of the model and actual class label. It is considered as trustworthy indicator, especially working with imbalanced dataset and calculated by following formula.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

True Positive Rate (TPR) It shows the proportion of actual positive sample that detected by the classifier as positive and calculate by Eq. (17).

$$TPR = \frac{TP}{TP + FN} \tag{17}$$

False Positive Rate (FPR) It is the ratio between the number of negative samples wrongly classified as positive and calculated by given formula (18).

$$FPR = \frac{FP}{FN + TN} \tag{18}$$

TPR vs FPR are plotted on a ROC curve at various classification thresholds. More samples are classified as positive when the classification threshold is lowered, which raises the number of both False positive and true positive. In binary classification, the thresholds are various probability thresholds that separate the two classes. It employs probability to calculate that how model performs at distinguishing between two classes.

- **Execution time:** The time take by the model to process the input data, perform classification or regression task.

9. Simulation setup and results discussion

9.1. Study configuration

The proposed framework was developed in Visual Studio Code using the Python programming language as delineated in Section 4. The entirety of our different four case studies are performed on windows 10 x64 with an Intel i5-8350U 1.90 GHz and 16 GB Ram. The electricity load profiles are obtained from SGCC. The load profiles are contains 42372 consumers in which 38757 are normal users and the rest of 3615 consumers labeled as theft represented in Table 2.

Table 4
Only subjected to preprocessed data.

Classifiers	Accuracy	AUC-Score	F1-Score	Precision	Recall	MCC	AUC-ROC	PR-AUC	Execution time
NB	76	**63**	26	18	47	17	62	34	1 s
LGBM	92	58	**27**	70	17	31	57	46	28 s
QDA	90	53	13	29	10	12	53	23	16 s
XGB	92	55	17	67	18	31	54	46	50 s
ADB	91	54	14	45	08	16	53	31	11 min 32 s
Proposed	**92**	58	26	**72**	18	31	**57**	**47**	8 min 37 s

Table 5
Only experiencing PCA for feature engineering.

Classifiers	Accuracy	AUC-Score	F1-Score	Precision	Recall	MCC	AUC-ROC	PR-AUC	Execution time
NB	82	62	27	21	37	19	61	32	1 s
LGBM	92	54	16	64	10	22	54	41	7 s
QDA	82	64	30	23	42	22	64	35	3 s
XGB	92	54	14	71	10	23	53	38	16 s
ADB	91	54	16	52	9	18	53	33	3 m 25 s
Proposed	**92**	**55**	20	63	11	**23**	**57**	40	3 m 1 s

9.1.1. Impact of preprocessed data on proposed framework

Adequate performance metrics for the testing of the proposed model are most important. However, the evaluation of a framework in terms of accuracy cannot provide a real assessment. The most suitable performance metric for Electricity Theft Detection (ETD) is Area Under the Curve (AUC) while dealing with imbalance dataset (Bradley, 1997). This shows the actual score of the positive class, which is most concerned with electricity utility. It represents the connection between True Positive Rate (TPR) also known as sensitivity and the rate of false positives (specificity) for samples detected by proposed approach. The classifier with a higher AUC has greater ability to distinguish between theft and normal class.

In this experiment, only preprocessed data is fed to the framework and experiment is performed as presents in Fig. 2 and Section 4 . As no feature engineering and oversampling method is applied on dataset, therefore results are not considerable. We can see in Table 4, accuracy is slightly high for all classifiers including the proposed except NB. However, the values of precision, recall, F1-score, and MCC for both the individual classifiers and the proposed model presents unsatisfactory results

Moreover, the proposed system slightly performs better in term of precision, AUC-ROC and PR-AUC as compared to the base level classifiers but the values are still unsatisfactory. As the proposed model achieves only 57% AUC-ROC and 47% PR-AUC; which clearly exhibits the absence of not having balanced proportion of both classes. It takes 8 min and 37 s to process all data which shows that proposed scheme is facing computational problem and lack of no feature engineering. Fig. 7(c) depicts the PR-AUC curve for proposed framework. The findings in Table 4 reveal that the proposed framework, along with all individual classifiers, gives a marginally higher level of accuracy. However, working with imbalanced dataset, accuracy is not a evaluation metric as it only shows the accurately classified samples by the model. While working with electricity consumption data, it is important to detect the theft samples.

The confusion metric for preprocessed data is presented in Fig. 7(d), which shows the difference of suspicious users by integrating True Negative (TN), False Positive (FP), True Positive (TP), False Negative (FN). The lower TP rate clearly shows that proposed method is not accurately detecting the class of interest (theft class). In conclusion, we can understand that the model presents unsatisfactory results without feature engineering and data balancing. In contrast, as represented in Figs. 7(b) and 7(c), the AUC and PR-recall curves proved that the achieved outcomes are influenced by the imbalanced nature of the data. Additionally, Table 4 demonstrates that the proposed model's execution time is longer due to the absence of feature engineering.

We have calculated FPR, FDR, FNR and FOR values for proposed and base level classifier to get better understanding of the classifier interpretability can be seen in Fig. 7(e). The obtained high FNR and FDR values indicates that the proposed model incorrectly fails to detect positive classes as positive and often detects benign samples as legitimate samples respectively. While, the smaller values of FPR (0.01) and FOR (0.07) clearly indicates that model performs well in these aspects. While these values further confirmed our statement that model poorly performs on preprocessed data.

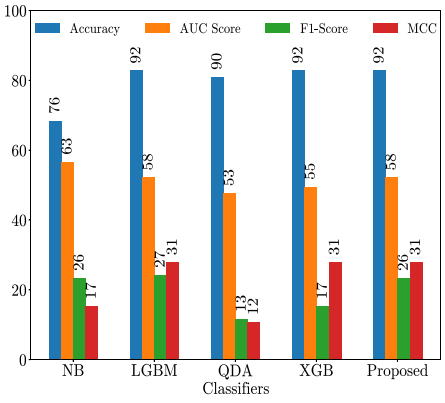
9.1.2. Effect of dimensionality reduction (DR) using PCA on proposed framework

In this case study, we exclusively employ PCA for feature extraction only to analyze the performance of our proposed framework on imbalanced dataset as no oversampling is performed in this experiment. The experimental results with only feature engineering are presented in Table 5.

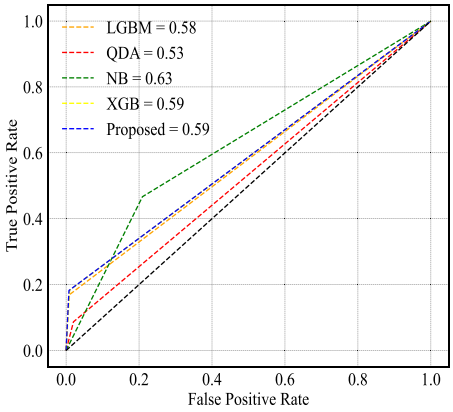
The simulations values are quite similar with slight change as compared with previous case study and can be observed in Table 4 and Fig. 8(a) due to only feature extraction performed. Nevertheless, the utilization of feature engineering has resulted in a reduction of almost 70% in the execution time compared to the previous experiment.

This advantage of feature engineering in resolving computational complexity is crucial as real-time data often consists of thousands of samples and redundant information. It takes 03 min and 01s to process all the data, which is highly lower than previous with 08 min 37s. We achieve this by incorporating PCA for dimensionality reduction in the dataset. It transform original features into new uncorrelated variables known as principle components. These mathematical principal components are ordered in terms of variance in highest to lowest order in the dataset.

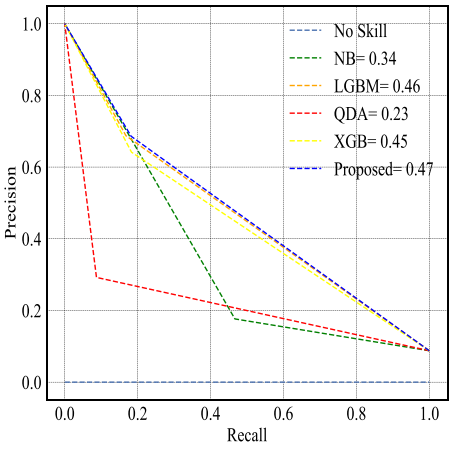
The performance graphs, AUC-ROC curve, and P-R curve of both the individual classifiers and meta classifier are presented in Figs. 8(a), 8(b) and 8(c). Furthermore, these outcomes are also biased and substantiated and can be verified by AUC-ROC and PR-Recall curves, as there is a huge class distribution issue due to absence of data balancing method. Moreover, the values in confusion matrix is slight better than previous case study. We can see a little increase in the values of FN, TN and FP. However, these values are still unacceptable and unsatisfactory. The Fig. 8(d) presents the values of confusion matrix for this experiment. Furthermore, the values of FPR, FDR, FNR and FOR values clearly verified that the performance of proposed model is almost similar with



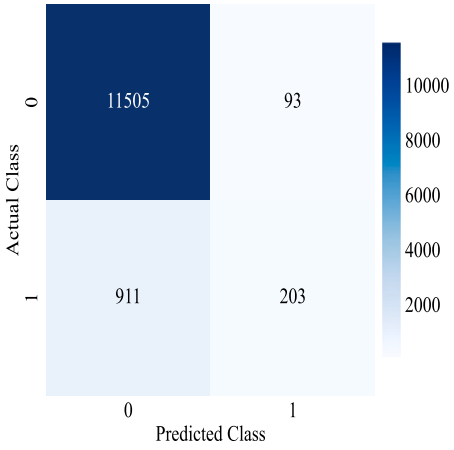
(a) Simulation results



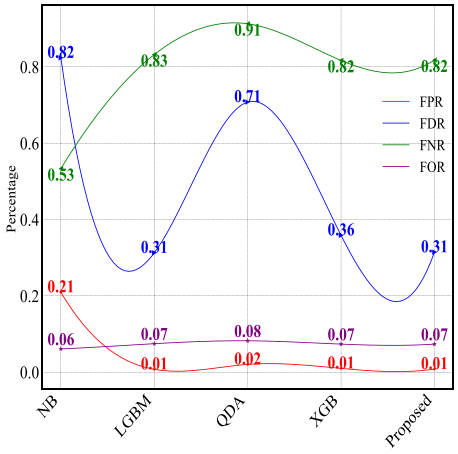
(b) AUC-ROC Curve



(c) P-R Curve



(d) Confusion matrix



(e) FNR, FPR, FDR, and FOR values of base and meta level

Fig. 7. Simulation results of preprocessed data on proposed framework.

previous case and proved that results are biased as no oversampling method is performed. Fig. 8(e) displays the curves for FPR, FDR, FNR, and FOR.

9.1.3. Effect of oversampling using borderline SMOTE on proposed framework

In this case, we employed Borderline SMOTE for balancing the majority and minority class samples. As Borderline SMOTE generate synthetic samples of minority class examples to counter the issue of class distribution. The findings presented in Table 6 and Fig. 9(a) provide a comprehensive comparison between the impact of data balancing and feature engineering.

The results exhibited in Fig. 9(a) demonstrate that the proposed model surpasses the base standalone classifiers in terms of accuracy 90%, AUC score 92%, F1-score 92%, and precision 92%. While these results obtained after applying borderline SMOTE on dataset to equal the class distribution between theft and suspicious consumers. Further, the

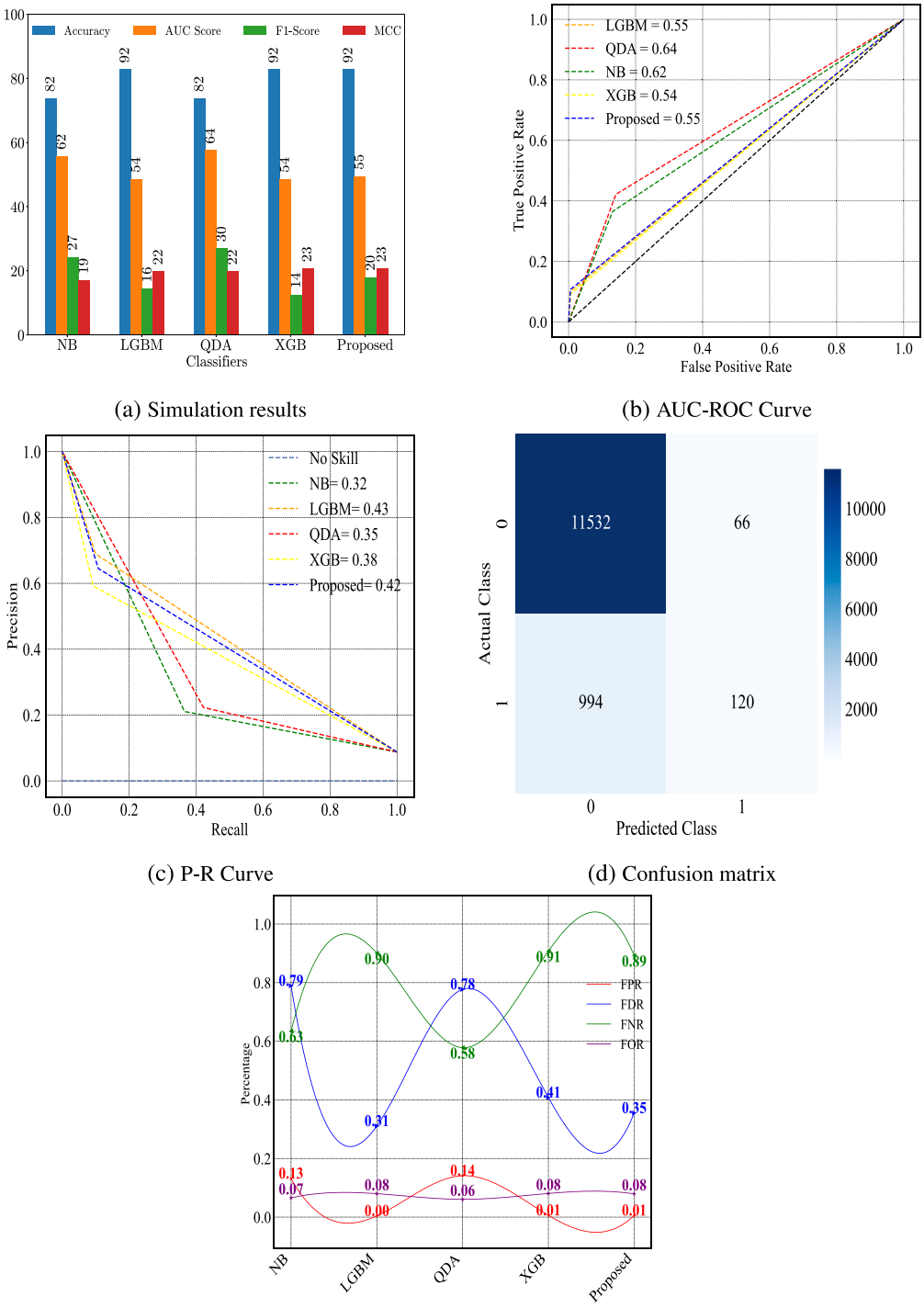


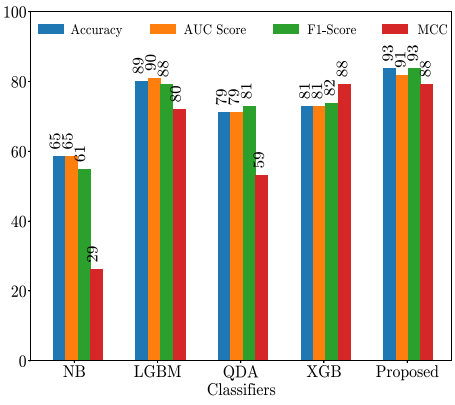
Fig. 8. Effect of dimensionality reduction (DR) using PCA on proposed framework.

model achieves 88% MCC score, which clearly indicates that classifier is accurately classifying the minority and majority class samples.

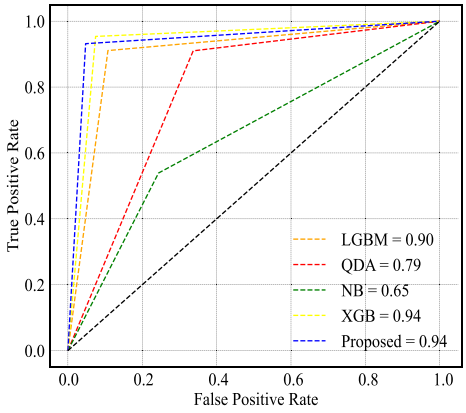
However, this improvement in performance comes at the cost of increased computational overhead, with the proposed approach requiring an additional 38 min to complete its execution, which is not considerable. The AUC-ROC and P-R curve illustrated in Figs. 9(b) and 9(c) respectively further confirm the superiority of our proposed model over the base classifiers. Furthermore, 9(d) illustrates the confusion matrix for proposed scheme. This matrix provides better results in terms of lower FN and high TP.

9.1.4. Effect of dimensionality reduction using PCA and oversampling using borderline SMOTE on proposed framework

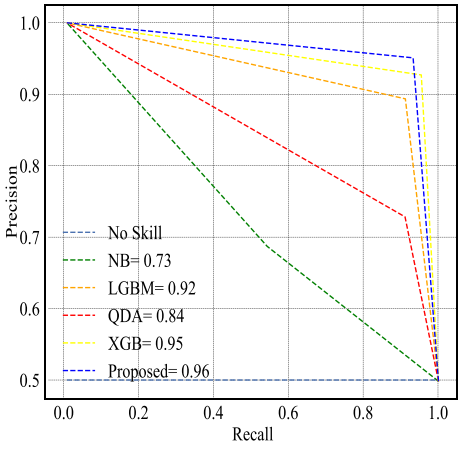
In this scenario, Borderline SMOTE and PCA is applied for oversampling and dimensionality reduction. The classification is then carried out at the base level using four individual classifiers and single classifier at meta level. Table 7 represents the simulation values of NB



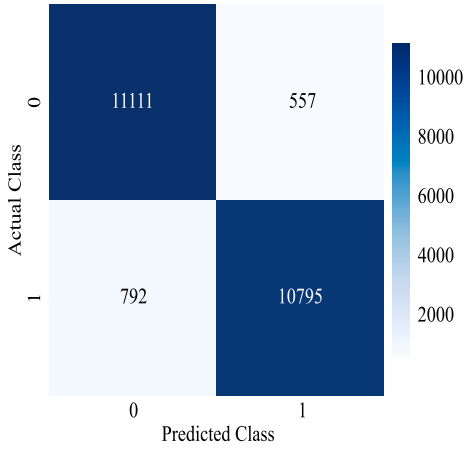
(a) Simulation results



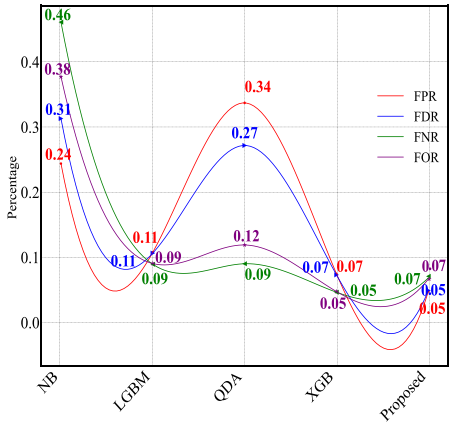
(b) AUC-ROC Curve



(c) P-R Curve



(d) Confusion matrix

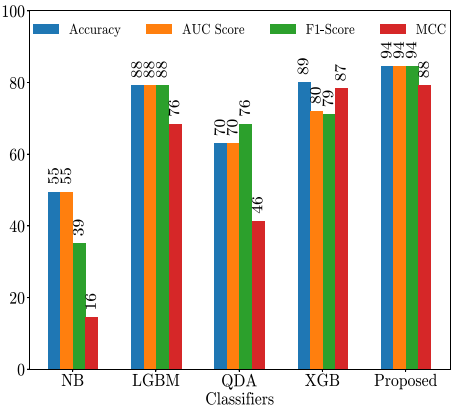


(e) FNR, FPR, FDR, and FOR values of base and meta level

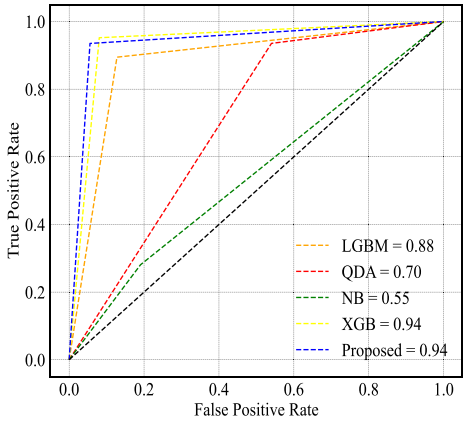
Fig. 9. Effect of oversampling using Borderline SMOTE on proposed framework.

techniques; while NB has a fast classification speed of one second for the entire dataset, it assumes that all attributes are independent of each other and do not contribute to each other. This is a limitation of NB for real-world cases, as such independence between attributes rarely happens in real-world cases. The simulation values in Table 7 indicate that NB has poor performance, achieving only 55% accuracy, 55% AUC score, 39% f1-score, 59% precision and only achieve 16% MCC.

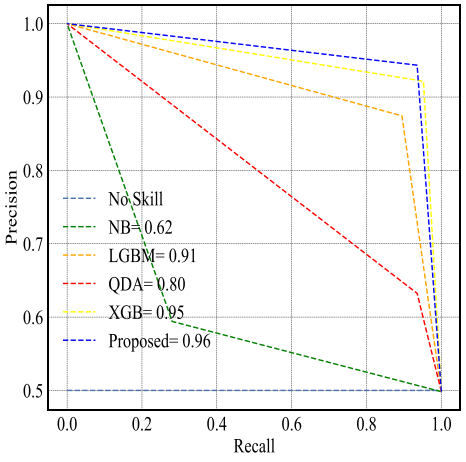
LGBM and XGB, on the other hand, offer good simulation results, as they are famous boosting method. Both techniques employ multiple weak learners to form a strong learner by reducing the gradient of the previous weak learner this slightly improves the results. However, LGBM splits the tree leaf-wise, which can lead to overfitting. These techniques achieve 88%, 89% accuracy, 88%, 80% AUC score, 88%, 79% F1-score, 87%, 80% precision, and 76%, 87% MCC respectively. QDA, the fourth base-level classifier used for first-stage classification, offers 70% accuracy, 70% AUC score, 76% f1-score, 63% precision and 46% MCC. However, in Fig. 10(a), QDA gives slightly better recall than the proposed, as the multivariate normal distribution of independent



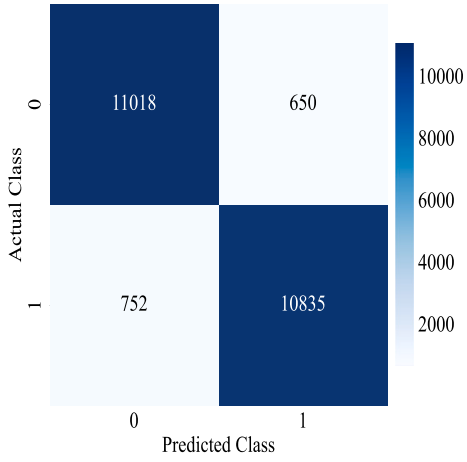
(a) Simulation results



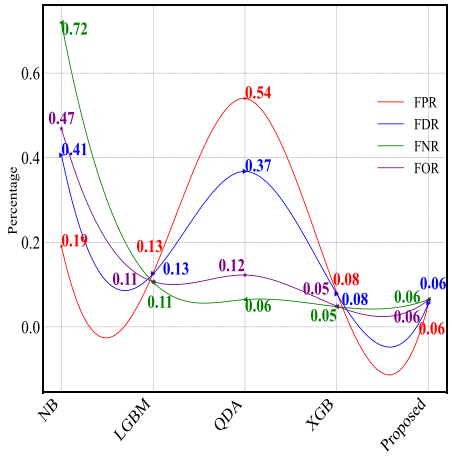
(b) AUC-ROC Curve



(c) P-R Curve



(d) Confusion matrix



(e) FNR, FPR, FDR, and FOR values of base and meta level

Fig. 10. Effect of dimensionality reduction using PCA and oversampling using Borderline SMOTE on proposed framework.

variables used as the classification rule for QDA leads to slightly better recall value.

Ada-boost classifier is then used for second-stage classification of the results obtained from base-level techniques. Considering accuracy, AUC score, f1-score, precision, and execution time, the proposed approach outperforms all base level techniques, achieving 90% accuracy, 90% AUC score, 89% F1-score, and 90% precision respectively. However, it takes 2 min and 35 s for procession the data, which is greater than all base level classifiers.

The poor performance of base-level classifiers can be attributed to several factors. NB assumes that all attributes are independent of each other, which is rarely the case in real-world scenarios. QDA, on the other hand, relies on the assumption of multivariate normal distribution of independent variables, which might not always hold true. LGBM, while offering good results, it splits the tree leaf-wise,

Table 6
Only subjected to over sampling using Borderline SMOTE.

Classifiers	Accuracy	AUC-Score	F1-Score	Precision	Recall	MCC	AUC-ROC	PR-AUC	Execution time
NB	65	65	61	69	54	29	64	72	3 s
LGBM	89	90	88	89	91	80	90	90	41 s
QDA	79	79	81	73	91	59	78	84	24 s
XGB	81	81	82	81	93	88	81	90	1 min 9 s
ADB	76	76	76	76	76	51	75	81	19 min 29 s
Proposed	**93**	**91**	**93**	**92**	**94**	**88**	**91**	**91**	13 min 4 s

Table 7
Experiencing borderline SMOTE oversampling and PCA dimensionality reduction.

Classifiers	Accuracy	AUC-Score	F1-Score	Precision	Recall	MCC	AUC-ROC	PR-AUC	Execution time
NB	55	55	39	59	28	16	54	61	1 s
LGBM	88	88	88	87	89	76	88	90	16 s
QDA	70	70	76	63	93	46	69	80	3 s
XGB	89	80	79	80	93	87	79	90	31 s
ADB	75	75	75	75	74	50	76	82	8 min 18 s
Proposed	94	94	94	95	94	88	92	92	4 min 30 s

leading to overfitting. Furthermore, XGB suffers with overfitting and model complexity, which leads to worse performance.

The superior performance of the meta-level classifier, on the other hand, can be attributed to its ability to combine the results of multiple base-level classifiers, effectively capturing the strengths of each classifier while mitigating their weaknesses.

The proposed framework achieved better results as compared to the base level classifiers. It achieves 93% accuracy, 92% AUC score, 95% precision and 88% MCC which is superior to all base classifiers and many existing approaches in literature review namely; wide and deep CNN (Zheng et al., 2018). Moreover, it takes 4 min and 30 s to process all data which remarkable achievement of the proposed scheme.

Additionally, working with electricity theft both AUC-ROC curve and PR-AUC curve is very crucial for power utility. The proposed model performs well in both AUC-ROC curve and PR-AUC curve, which is very important while working on ETD. The framework presents strong performance, as proved by a higher area under the curve for both AUC-ROC and PR-AUC curves. This exhibits that the framework effectively achieves TPR and maintains higher precision. It outperforms in term of AUC-ROC and PR-AUC scores for both with 92% and beats all other existing models presented in existing literature. Figs. 10(b) and 10(c) shows AUC-ROC and PR-ROC curves respectively.

Moreover, we can see the difference of positive predictions in Fig. 7(d), and Fig. 10(d) after and before applying the PCA and Borderline SMOTE on the dataset.

For this case, the proposed model performs relatively balanced performance for FPR, FNR, FDR and FOR. After mitigating class imbalance and improve the model ability to learn from the features have contributed to this balanced performance. As we can see in Fig. 10(e) model predict 6% FPR that clearly indicates that model classify 6% of benign samples as theft samples. The smaller FPR rate indicates the robustness of our proposed framework. Further, 6% of FDR clearly indicates that prediction of model is incorrect 6% of times. Similarly, 6% of FNR indicates that models predict 6% of theft samples as honest samples from all predictions and 6% of FOR rate represents that all negative predictions are actually positive. From all these results, it can be concluded that the performance of proposed stacked ensemble method were relatively high. This proves that the integration of oversampling method and feature engineering method contributes to improve the models interpretability and reduces simulation time.

In conclusion, feature engineering and data balancing technique are utilized on the dataset for proposed approach, which significantly improves the classification performance. The strengths and limitations of each classifier used in the study were discussed, and the importance of working with AUC-ROC and PR-AUC curves in power utility applications is highlighted.

10. Conclusion

In this research, we proposed stacked ensemble method for the detection of electricity theft to secure smart grid. In this framework, Borderline SMOTE is employed for data balancing and PCA for feature engineering. Furthermore, a stacked model with base layer and meta layer are proposed for the classification of theft and honest consumers. Four individual classifiers, NB, XGB, QDA and LGBM are used at base layer. A single Ada-boost is used at meta layer to further classification of the results at second stage. The proposed framework were evaluated in four different scenarios; in first, with only preprocessed data to observe the performance of the framework, in second, only feature engineering is performed on the dataset to observe the credibility of the framework in the absence of data balancing method. In third case, only data balancing with BDSmote is performed on the dataset to observe the simulation values of the method, and lastly both feature engineering and data balancing is performed on the preprocessed dataset to observe the experimental values.

CRediT authorship contribution statement

Muhammad Hashim: Writing – review & editing, Writing – original draft, Software, Visualization,Resources, Methodology, Conceptualization. **Laiq Khan:** Writing – review & editing, Supervision, Project administration, Formal analysis, Data curation, Conceptualization. **Nadeem Javaid:** Writing – review & editing, Supervision, Investigation, Formal analysis, Data curation, Conceptualization. **Zahid Ullah:** Writing – review & editing, Writing – original draft, Resources, Project administration, Conceptualization. **Aymin Javed:** Formal analysis, Data curation.

Declaration of competing 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

Data will be made available on request.

References

Al-Turjman, Fadi, Abujubbeh, Mohammad, 2019. IoT-enabled smart grid via SM: An overview. Future Gener. Comput. Syst. 96, 579–590.

Aldegheishem, A., Anwar, M., Javaid, N., Alrajeh, N., Shafiq, M., Ahmed, H., 2021. Towards sustainable energy efficiency with intelligent electricity theft detection in smart grids emphasising enhanced neural networks. IEEE Access 9, 25036–25061.

Almazroi, A.A., Ayub, N., 2021. A novel method CNN-LSTM ensembler based on black widow and blue monkey optimizer for electricity theft detection. IEEE Access 9, 141154-141166.

Anagnostopoulos, C., Tasoulis, D.K., Adams, N.M., Pavlidis, N.G., Hand, D.J., 2012. Online linear and quadratic discriminant analysis with adaptive forgetting for streaming classification. Stat. Anal. Data Min.: ASA Data Sci. J. 5 (2), 139–166.

Anon, 2023a. https://math.stackexchange.com/questions/3778238/understanding-adaboost-algorithm. (Last accssessed 17 May 2023).

Anon, 2023b. XGBoost algorithm: Long may she reign!—Towards data science. [online]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.

Bizzozero, Federica, Gruosso, Giambattista, Vezzini, Nicolò, 2016. A time-of-use-based residential electricity demand model for smart grid applications. In: 2016 IEEE 16th International Conference on Environment and Electrical Engineering. EEEIC, IEEE.

Bradley, Andrew P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. 30 (7), 1145–1159.

Brownlee, J., 2021. How to Develop a Light Gradient Boosted Machine (LightGBM) Ensemble. MachineLearningMastery.com, [Online]. Available: https://machinelearningmastery.com/light-gradient-boosted-machine-lightgbmensemble. (Accessed 01 April 2023).

Buzau, M.M., Tejedor-Aguilera, J., Cruz-Romero, P., Gómez-Expósito, A., 2020. Hybrid deep neural networks for detection of non-technical losses in electricity smart meters. IEEE Trans. Power Syst. 35 (2), 1254–1263. http://dx.doi.org/10.1109/TPWRS.2019.2943115.

Choi, Deok-Kee, 2019. Data-driven materials modeling with xgboost algorithm and statistical inference analysis for prediction of fatigue strength of steels. Int. J. Precis. Eng. Manuf. 20, 129–138.

Cui, X., Liu, S., Lin, Z., Ma, J., Wen, F., Ding, Y., et al., 2021. Two-step electricity theft detection strategy considering economic return based on convolutional autoencoder and improved regression algorithm. IEEE Trans. Power Syst. 37 (3), 2346–2359.

Džeroski, S., Ženko, B., 2004. Is combining classifiers with stacking better than selecting the best one? Mach. Learn. 54 (3), 255–273.

Gunturi, S.K., Sarkar, D., 2021. Ensemble machine learning models for the detection of energy theft. Electr. Power Syst. Res. 192, 106904. http://dx.doi.org/10.1016/J.EPSR.2020.106904.

Gupta, Amit, Lohani, M.C., Manchanda, Mahesh, 2021. Financial fraud detection using naive bayes algorithm in highly imbalance data set. J. Discrete Math. Sci. Cryptogr. 24 (5), 1559–1572.

Han, Hui, Wang, Wen-Yuan, Mao, Bing-Huan, 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August (2005) 23-26, Proceedings, Part I 1. Springer Berlin Heidelberg.

Hu, Chengming, Yan, Jun, Wang, Chun, 2019. Advanced cyber–physical attack classification with extreme gradient boosting for smart transmission grids. In: 2019 IEEE Power and Energy Society General Meeting. PESGM, IEEE.

Huang, Y., Xu, Q., 2021. Electricity theft detection based on stacked sparse denoising autoencoder. Int. J. Electr. Power Energy Syst. 125, 106448.

Iwashita, A.S., Rodrigues, D., Gastaldello, D.S., de Souza, A.N., Papa, J.P., 2021. An incremental optimum-path forest classifier and its application to non-technical losses identification. Comput. Electr. Eng. 95, 107389.

Jindal, A., Dua, A., Kaur, K., Singh, M., Kumar, N., Mishra, S., 2016. Decision tree and SVM-based data analytics for theft detection in smart grid. IEEE Trans. Ind. Inf. 12 (3), 1005–1016, %@ 1551-3203.

Jokar, Paria, Arianpoo, Nasim, Leung, Victor C.M., 2015. Electricity theft detection in AMI using customers' consumption patterns. IEEE Trans. Smart Grid 7 (1), 216–226.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., et al., 2017. LightGBM: A highly efficient gradient boosting decision tree. Proc. Adv. Neural Inf. Process. Syst. 30, 1–9.

Lee, J., Sun, Y.G., Sim, I., Kim, S.H., Kim, D.I., Kim, J.Y., 2022. Non-technical loss detection using deep reinforcement learning for feature cost efficiency and imbalanced dataset. IEEE Access 10, 27084–27095.

Li, S., Han, Y., Yao, X., Yingchen, S., Wang, J., Zhao, Q., 2019. Electricity theft detection in power grids with deep learning and random forests. J. Electr. Comput. Eng. 2019.

Lin, G., Feng, H., Feng, X., Wen, H., Li, Y., Hong, S., Ni, Z., 2021. Electricity theft detection in power consumption data based on adaptive tuning recurrent neural network. Front. Energy Res. 9, 773805.

Mabunga, Z., Cruz, J.D., Magwili, G., Samortin, A., 2020. Development of sanitary landfill's groundwater contamination detection model based on machine learning algorithms. In: 2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management. HNICEM, IEEE, pp. 1–6.

Musleh, Ahmed S., et al., 2019. Detection of false data injection attacks in smart grids: A real-time principle component analysis. In: IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society, Vol. 1. IEEE.

Nawaz, A., Ali, T., Mustafa, G., Rehman, S.U., Rashid, M.R., 2023. A novel technique for detecting electricity theft in secure smart grids using CNN and XG-boost. Intell. Syst. Appl. 17, 200168.

Nazmul Hasan, M., Toma, R.N., al Nahid, A., Manjurul Islam, M.M., Kim, J.M., 2019. Electricity theft detection in smart grid systems: A CNN-LSTM based approach. Energies 12 (17), 3310.

Nes, 2020. Energy theft and fraud reduction, Smart Energy International. Available at: https://www.smart-energy.com/industry-sectors/energy-grid-management/energy-theft-and-fraud-reduction/. (Accessed 9 February 2023).

Northeast Group, L.L.C., 2018. $96 Billion is lost every year to electricity theft. Available at: https://www.prnewswire.com/news-releases/96-billion-is-lost-every-year-to-electricity-theft-300453411.html. (Accessed 9 February 2023).

Palahalli, Harshavardhan, Ragaini, Enrico, Gruosso, Giambattista, 2019. Smart grid simulation including communication network: A hardware in the loop approach. IEEE Access 7, 90171–90179.

Qu, Zhijian, et al., 2021. A combined genetic optimization with AdaBoost ensemble model for anomaly detection in buildings electricity consumption. Energy Build. 248, 111193.

Rajiv, Punmiya., Choe, Sangho., 2019. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. IEEE Trans. Smart Grid 10 (2), 2326–2329.

Ramos, C.C., Rodrigues, D., de Souza, A.N., Papa, J.P., 2016. On the study of commercial losses in Brazil: A binary black hole algorithm for theft characterization. IEEE Trans. Smart Grid 9 (2), 676–683.

Shehzad, Faisal, et al., 2023. Deep learning-based meta-learner strategy for electricity theft detection. Front. Energy Res. 11, 1–13.

Shi, J., Gao, Y., Gu, D., Li, Y., Chen, K., 2023. A novel approach to detect electricity theft based on conv-attentional transformer neural network. Int. J. Electr. Power Energy Syst. 145, 108642.

Shukla, A., Dutta, S., Sahu, S.K., Sadhu, P.K., 2023. A narrative perspective of island detection methods under the lens of cyber-attack in data-driven smart grid. J. Electr. Syst. Inf. Technol. 10 (1), 1–32.

Taha, A.A., Malebary, S.J., 2020. An intelligent approach to credit card fraud detection using an optimized light gradient boosting machine. IEEE Access 8, 25579–25587.

Tang, J., Alelyani, S., Liu, H., 2014. Data Classification: Algorithms and Applications. In: Data Mining and Knowledge Discovery Series, CRC Press, Boca Raton, FL, USA, pp. 37–64.

Tehrani, Soroush Omidvar, Shahrestani, Afshin, Yaghmaee, Mohammad Hossein, 2022. Online electricity theft detection framework for large-scale smart grid data. Electr. Power Syst. Res. 208, 107895.

Ting, Kai Ming, Witten, Ian H., 1997. Witten stacked generalization: when does it work? pp. 866–871.

Xia, R., Gao, Y., Zhu, Y., Gu, D., Wang, J., 2022. An efficient method combined data-driven for detecting electricity theft with stacking structure based on grey relation analysis. Energies 15 (19), 7423.

Yan, Zhongzong, Wen, He, 2021. Performance analysis of electricity theft detection for the smart grid: An overview. IEEE Trans. Instrum. Meas. 71, 1–28.

Yao, R., Wang, N., Liu, Z., Chen, P., Ma, D., Sheng, X., 2021. Intrusion detection system in the smart distribution network: A feature engineering based AE-LightGBM approach. Energy Rep. 7, 353–361.

Yeckle, Jaime, Tang, Bo, 2018. Detection of electricity theft in customer consumption using outlier detection algorithms. In: 2018 1st International Conference on Data Intelligence and Security. ICDIS, IEEE.

Zeng, K., Liu, J., Wang, H., Zhao, Z., Wen, C., 2020. Research on adaptive selection algorithm for multi-model load forecasting based on adaboost. In: IOP Conference Series: Earth and Environmental Science, Vol. 610, No. 1. IOP Publishing, 012005.

Zhang, L., Wen, J., Li, Y., Chen, J., Ye, Y., Fu, Y., Livingood, W., 2021. A review of machine learning in building load prediction. Appl. Energy 285, 116452.

Zheng, Z., Yang, Y., Niu, X., Dai, H.N., Zhou, Y., 2018. Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. IEEE Trans. Ind. Inform. 14 (4), 1606–1615. http://dx.doi.org/10.1109/TII.2017.2785963.

Zidi, S., Mihoub, A., Mian Qaisar, S., Krichen, M., Abu Al-Haija, Q., 2023. Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment. J. King Saud Univ. - Comput. Inf. Sci. 35 (1), 13–25. http://dx.doi.org/10.1016/j.jksuci.2022.05.007.