

Article

Electricity Theft Detection Using Supervised Learning Techniques on Smart Meter Data

Zahoor Ali Khan ^{1,*}, Muhammad Adil ², Nadeem Javaid ², Malik Najmus Saqib ³,
Muhammad Shafiq ⁴ and Jin-Ghoo Choi ^{4,*}

¹ Computer Information Science, Higher Colleges of Technology, Fujairah 4114, UAE

² Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad 44000, Pakistan; adilxyz1@gmail.com (M.A.); nadeemjavaidqau@gmail.com (N.J.)

³ Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, Jeddah 21959, Saudi Arabia; mnajam@uj.edu.sa

⁴ Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, Gyeongbuk 38541, Korea; shafiq@ynu.ac.kr

* Correspondence: zkhan1@hct.ac.ae (Z.A.K.); jchoi@yu.ac.kr (J.-G.C.)

Received: 10 August 2020; Accepted: 23 September 2020; Published: 28 September 2020

Abstract: Due to the increase in the number of electricity thieves, the electric utilities are facing problems in providing electricity to their consumers in an efficient way. An accurate Electricity Theft Detection (ETD) is quite challenging due to the inaccurate classification on the imbalance electricity consumption data, the overfitting issues and the High False Positive Rate (FPR) of the existing techniques. Therefore, intensified research is needed to accurately detect the electricity thieves and to recover a huge revenue loss for utility companies. To address the above limitations, this paper presents a new model, which is based on the supervised machine learning techniques and real electricity consumption data. Initially, the electricity data are pre-processed using interpolation, three sigma rule and normalization methods. Since the distribution of labels in the electricity consumption data is imbalanced, an Adasyn algorithm is utilized to address this class imbalance problem. It is used to achieve two objectives. Firstly, it intelligently increases the minority class samples in the data. Secondly, it prevents the model from being biased towards the majority class samples. Afterwards, the balanced data are fed into a Visual Geometry Group (VGG-16) module to detect abnormal patterns in electricity consumption. Finally, a Firefly Algorithm based Extreme Gradient Boosting (FA-XGBoost) technique is exploited for classification. The simulations are conducted to show the performance of our proposed model. Moreover, the state-of-the-art methods are also implemented for comparative analysis, i.e., Support Vector Machine (SVM), Convolution Neural Network (CNN), and Logistic Regression (LR). For validation, precision, recall, F1-score, Matthews Correlation Coefficient (MCC), Receiving Operating Characteristics Area Under Curve (ROC-AUC), and Precision Recall Area Under Curve (PR-AUC) metrics are used. Firstly, the simulation results show that the proposed Adasyn method has improved the performance of FA-XGBoost classifier, which has achieved F1-score, precision, and recall of 93.7%, 92.6%, and 97%, respectively. Secondly, the VGG-16 module achieved a higher generalized performance by securing accuracy of 87.2% and 83.5% on training and testing data, respectively. Thirdly, the proposed FA-XGBoost has correctly identified actual electricity thieves, i.e., recall of 97%. Moreover, our model is superior to the other state-of-the-art models in terms of handling the large time series data and accurate classification. These models can be efficiently applied by the utility companies using the real electricity consumption data to identify the electricity thieves and overcome the major revenue losses in power sector.

Keywords: data pre-processing; electricity theft; imbalance data; parameter tuning; smart grid

1. Introduction

1.1. Background and Motivation

The smart grid system is defined as the conventional electricity network with the addition of digital communication technologies, i.e., sensors and smart meters. Recent studies in [1–4] show that the smart grid can help in efficient management of electrical power. The transactive energy framework [5] and short term load scheduling [6] are introduced to ensure optimal use of installed resources in the smart grid system. The hierarchical energy management system is presented in [7] to reduce the peak hours and trade more electricity at lower prices. The information-gap decision theory based solution is utilized to reduce the intermittent nature of renewable energies [8]. In a smart grid, the smart meter exchanges the information between electricity users and the grid. It records a huge amount of data, including the electrical energy consumption of consumers. Exploiting these data, the artificial intelligence techniques can track the energy consumption patterns of consumers and accurately identify the electricity thieves.

The electricity thieves bring major revenue losses to the electric utility. Electricity losses in transmission and distribution can be generally categorized into Technical Losses (TL) and Non-Technical Losses (NTL). TL occurs due to power dissipation in overhead power lines, transformers, and other substation equipment that are used to transfer electricity. NTL primarily consists of electricity theft. Electricity theft is defined as the energy consumed without authorization of power utility [9]. It includes bypassing the electricity meter, energy corruption of unregistered connections, tampering the meter reading, and direct hooking [10]. It is accountable for major revenue losses and decreases power quality [11]. A recent survey estimates that every year the power utility companies lose more than \$20 billion worldwide [12]. The NTL affects both the developed and developing countries. For instance, in Pakistan, the electricity transmission and distribution losses of 17.5% were recorded for the years 2017–2018 [13]. India also loses about \$4.5 billion each year due to electricity theft. A recent survey estimates that 20% of the total electricity is lost in India due to the illegal electricity consumption [14]. This problem also affects the rich nations. In the United States, the losses due to illegal electricity consumption are about \$6 billion annually, while, in the UK, the power losses exceed up to £175 million every year [15]. Moreover, electricity theft behaviors can also affect the operations and reliability of the power system. It decreases the power quality by overloading of transformers and voltage imbalances.

1.2. Literature Review

The researchers have recently implemented various approaches to detect the electricity theft. These approaches can be divided into three categories: state based solutions, game theory, and machine learning. The state based solutions use the additional hardware equipment like wireless sensors, distribution transformers, and smart meters to detect the electricity theft [16]. This method has a high cost of implementation due to the need of additional hardware equipment. In a game theory based method, there is assumed to be a game between the power utility and the electricity thieves. The outcome of a game can be derived from the difference between the electricity consumption behavior of electricity thieves and benign users [17]. However, it needs to define a utility function for all the players in a game, which is quite challenging. The machine learning techniques are widely used for ETD. They can be further categorized into unsupervised techniques (clustering) and supervised techniques (classification) that are later applied to unlabelled datasets in order to classify fraudulent and normal consumers. The existing methods used for ETD are presented in Table 1, which contains their contributions and limitations.

Table 1. Existing methods used for ETD.

Methods	Contributions	Limitations
Hardware based [16]	Its focus is on designing specific hardware devices in order to detect electricity theft	High cost of hardware installation
Game theory [17]	There is a game between the electricity thieves and the utility. The outcome of a game can be derived from the difference between the electricity consumption behavior of electricity thieves and benign users	This method needs to define a utility function for all the players in a game, which is quite challenging.
Machine learning [18–25]	It uses the smart meter data to effectively detect anomalous consumption behavior of dishonest consumers	Performance is poor on highly imbalanced data

Positioning of Our Work in the Literature

Our approach proposes a solution based on supervised learning. Therefore, we will study the details about the recent advances made in supervised learning techniques. The Support Vector Machine (SVM) and Logistic Regression (LR) are mostly used for ETD [18]. These techniques perform better when the dataset is small. However, these techniques are not effective when the dataset is large and extremely imbalanced. Hasan et al. [19] proposed a hybrid model consisting of CNN and Long Short Term Memory (LSTM). The CNN is utilized for feature extraction while LSTM used the refined features to classify the data into honest consumers and electricity thieves. To solve the problem of an imbalanced dataset, the Synthetic Minority Over Sampling Technique (SMOTE) is utilized. It [18] has achieved good results. i.e., precision 90% and recall 87%. However, the overfitting problem is not considered, which is caused by the addition of duplicate information through SMOTE. In [16], the authors proposed a hybrid model based on Multi-Layer Perceptron (MLP) and LSTM for ETD. This model detects the NTL by combining the auxiliary data through MLP and electricity consumption data through LSTM. However, the unbalanced data problem is not solved before classification. Moreover, the FPR of this model is high due to training on less data. It has achieved 54.5% PR-AUC, when 80% data are used for training. In [20], the authors addressed the issue of NTL detection using a Maximum Overlap Decomposition and Packet Transform (MODWPT) and Random Under Sampling Boosting (RUSBoost) techniques. The RUSBoost method is effective in handling the imbalanced data. However, the authors do not perform optimization to select best parameter values to improve the classification process. Moreover, the random under sampling technique reduces the data size and results in under fitting the model. To address the issue of power losses in Brazil, Ramos et al. [21] designed a Binary Black Hole Algorithm (BBHA) for NTL detection in Brazil. The accuracy comparison shows that BBHA outperforms other optimization techniques, i.e., Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). However, no reliable evaluation metrics like precision and recall are used to validate the performance of the system. The reliable evaluation is very necessary in case of imbalanced binary classification problems.

Authors in [18] proposed a solution based on XGBoost and SVM for the detection of NTL in the smart grid. The aim of this study is to rank the list of consumers based on the smart meter data and extract features from the auxiliary dataset. The XGBoost is utilized that operates as an ensemble model and boosts the classification performance. However, the data pre-processing is not considered to refine the input data. The performance of machine learning algorithms is dependent on the quality of input data. In [22], the authors proposed a new technique to detect the NTL, which is based on Maximum Information Coefficient (MIC) and Fast Search by Finding of Density Peaks (FSFDP). The refined data are achieved by the MIC method, while FSFDP is used for classification. However, it needs an

additional cost of hardware installation. The summary of existing work related to supervised learning techniques is given in Table 2. It gives the information about contributions and limitations of the existing work done in ETD using the supervised learning techniques.

Table 2. Performance of supervised machine learning techniques in the literature. MODWPT: Maximum Overlap Packet Transform; LSTM: Long Short Term Memory; CNN: Convolutional Neural Network; XGBoost: Extreme Gradient Boosting Technique; SGCC: State Grid Corporation of China; MLP: Multi Layer Perceptron; SMOTE: Synthetic Minority Over-sampling Technique; SVM: Support Vector Machine; BBHA: Binary Black Hole Algorithm.

Dataset	Supervised Techniques	Data Balancing	Contributions	Limitations
SGCC [19]	LSTM, CNN	SMOTE	The CNN is utilized for feature extraction, while LSTM uses the refined features to classify the data into honest consumers and electricity thieves.	The overfitting problem is not considered, which is caused by the addition of duplicate information through SMOTE
Endesa [16]	MLP and LSTM	Not handled	Detect the NTL by combining the auxiliary data through MLP and electricity consumption data through LSTM	The imbalanced data are not balanced before classification
Honduras [20]	MODWPT, RUSBoost	National grid of Brazil	The MODWPT gives the refined input and RUSBoost method balances the labels in the data before classification	The random under sampling technique reduces the data size and results in underfitting the model
Brazilian utility [21]	BBHA	Not handled	Use of binary black hole optimization technique to identify the NTL	No reliable evaluation is performed to validate the performance of the system
Endesa [18]	SVM, XGBoost	RUS	The XGBoost is utilized that operates as an ensemble method and boosts the classification performance	The data pre-processing is not considered to refine the input data
Irish data [22]	MIC, FSFD	Not handled	The refined data are achieved by MIC method, while FSFD is used for classification.	This model has a high cost of hardware installation
NAB [23]	LSTM-GMM	Not handled	The authors enhanced the internal structure of LSTM to solve the gradient vanishing problem	The model is complex and its execution time is high
EISA [24]	CNN, RF	SMOTE	The generalized performance is achieved by using the decision trees along with CNN	The SMOTE generate synthetic data, which causes overfitting issues

Ding et al. [23] solve the gradient vanishing problem by enhancing the internal structure of LSTM to detect the electricity theft. This approach is based on LSTM and Gaussian Mixture Model (GMM). This model achieved excellent results. i.e., precision 90.1% and recall 91.9%. However, this model has high execution time. In [24], the authors utilize the CNN model for detecting the electricity theft. In [18] CNN, the classification through fully connected layers leads towards the degradation of generalization. Therefore, the authors used Random Forest (RF) for final classification. Moreover, the imbalanced data are handled using SMOTE. The generalized performance is achieved by using the decision trees along with CNN. However, the SMOTE generates synthetic data, which causes the overfitting problem. Authors in [25] used a gradient Boosting theft detector for NTL detection. This technique improves its performance by learning from an ensemble of decision trees, which shows the effectiveness of the model. The simulation result shows that a gradient Boosting theft detector is superior to other machine learning techniques.

The performances of the existing Electricity Theft Detection (ETD) methods are reasonable. However, these methods have some limitations, which are given below.

1. Conventional ETD includes the manual methods, i.e., humanly checking the meter readings and examining the direct hooking of power transmission lines. However, these methods require the additional cost for hiring the inspection teams.
2. The game theory based solutions have a low detection rate and high False Positive Rate (FPR) [26].
3. The state based solution is expensive because it requires an additional cost for hardware implementation [27].
4. The major problem in ETD using machine learning techniques is handling the unbalanced data. In traditional models, this problem is left untreated. Some authors (as mentioned in Table 2) use the RUS and SMOTE methods, which cause the loss of information and overfitting problem, respectively.
5. In most cases, the available data contain erroneous values, which reduce the classification accuracy [28].
6. The traditional machine learning techniques like Logistic regression (LR) and Support Vector Machine (SVM) have poor classification performance for massive data [28].

1.3. Contributions

The flowchart of proposed methodology for ETD is also given in Figure 1. The mapping of problems addressed and our proposed approach is given in Table 3. In the proposed methodology, the electricity data are pre-processed using interpolation of missing values, three sigma rule and normalization methods to compute the missing values and remove the outliers in the data. An Adasyn algorithm is proposed for handling the imbalanced dataset. Afterwards, the balanced data are fed into the Visual Geometry Group (VGG-16) module for features extraction. A VGG-16 detects abnormal patterns in electricity consumption data. Finally, the extracted features are passed to the Firefly Algorithm based Extreme Gradient Boosting (FA-XGBoost) module for classification. The main applications of this paper are listed below.

- The proposed approach provides the solution for the problem present in the power sector, such as to wastage of electrical power due to electricity theft.
- This model can efficiently be applied by the utility companies using the real electricity consumption data to identify the electricity thieves and reduce the energy wastage.
- The proposed approach can be used against the all types of consumers who steal the electricity.

The key contributions of this paper are:

- A comprehensive data pre-processing is performed using interpolation, three sigma rule, and normalization methods to deal with missing values and outliers in the dataset. The data pre-processing step gives the refined input, which improves the performance of the classifier.

- A class balancing technique, Adasyn, is proposed to address the problem of imbalance data. The benefit of using Adasyn is two-fold. Firstly, it improves the learning performance of classifier to be more focused on theft cases that are harder to learn. Secondly, it prevents the model from being biased.
- We have introduced a new technique VGG-16 to solve the problem of overfitting to improve the classification performance. This technique is never being used before in ETD domain, and it has improved the accuracy of the classification model. The VGG-16 efficiently extracts useful information from data to truly represent electricity theft cases.
- XGBoost is applied to predict final classification, which improves the performance by combining multiple weak learners to make a strong learner.
- Along with XGBoost, an optimization technique, the Firefly Algorithm (FA) is utilized for efficient parameter optimization of the classifier.
- We conduct extensive simulations on real electricity consumption data set and for comparative analysis, precision, recall, F1-score, Matthews Correlation Coefficient (MCC), Receiving Operating Characteristics Area Under Curve (ROC-AUC), and Precision Recall Area Under Curve (PR-AUC) are used as performance metrics.

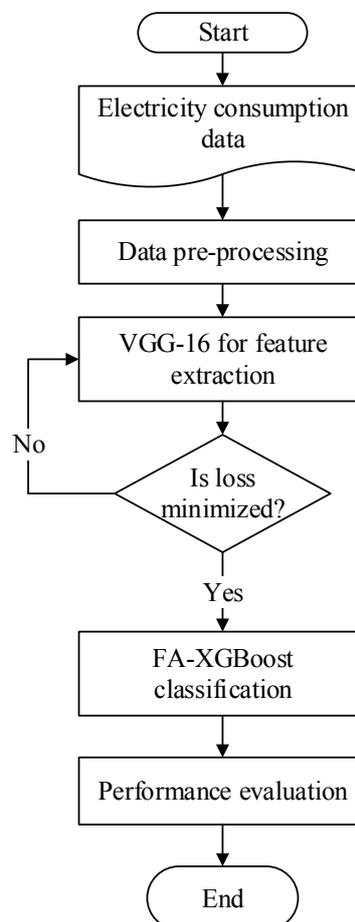


Figure 1. Flowchart of the proposed model for ETD.

Table 3. Mapping of problems addressed and proposed solution.

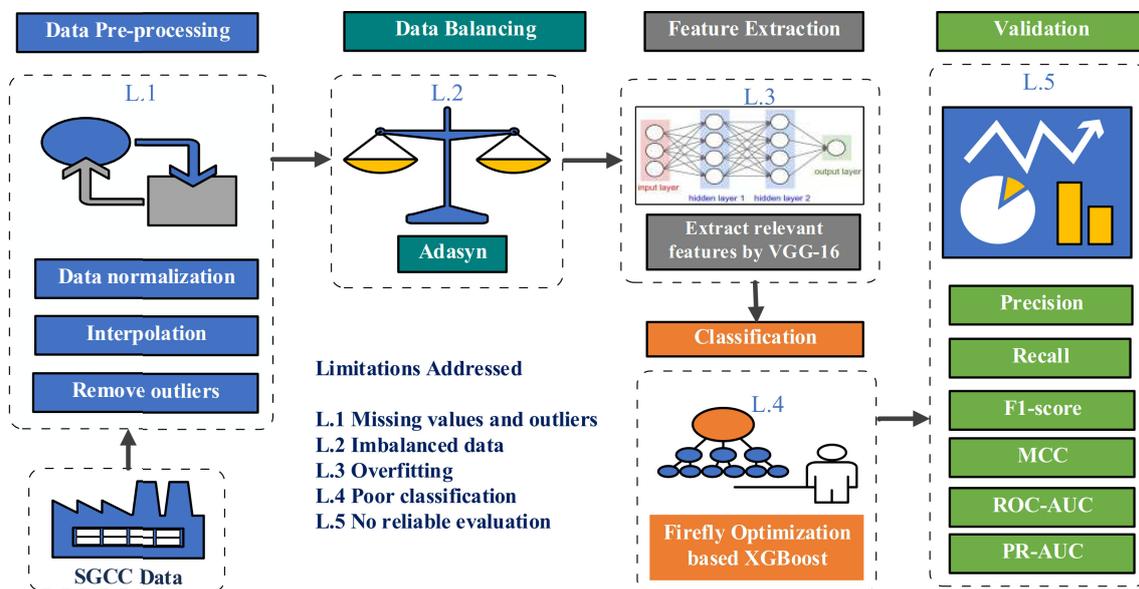
Limitation Number	Limitation Identified	Solution Number	Proposed Solution
L.1	Missing values and outliers	S.1	Pre-processing
L.2	Imbalanced data	S.2	Adasyn
L.3	Overfitting	S.3	VGG-16
L.4	Weak classification	S.4	FA-XGBoost
L.5	Reliable Evaluation	S.5	Precision, Recall, F1-Score, MCC, ROC-AUC, PR-AUC

1.4. Organization of Paper

The remaining paper is categorized as follows. Section 2 shows the proposed methodology. Section 3 provides the simulation results. Finally, this paper is concluded in Section 4.

2. Proposed System Model

Our proposed solution for ETD is presented in Figure 2. The proposed system model mainly consists of five parts: data pre-processing, data balancing, feature extraction, classification, and validation. Initially, electricity data are pre-processed using interpolation of missing values, three sigma rule, and normalization methods. Secondly, the pre-processed data are passed to the next model for data balancing. An Adasyn algorithm is used to balance the data. Thirdly, a VGG-16 is used to extract the important features from time series data and finally the important features are given to FA-XGBoost for classification. For comparative analysis, we use various performance metrics, i.e., precision, recall, F1-score, ROC-AUC, and PR-AUC to validate the effectiveness of our proposed model.

**Figure 2.** Proposed system model.

2.1. Overview of Proposed Methodology

The proposed methodology for ETD is described in the following subsections.

2.1.1. Information of Collected Data

The proposed system is tested using a high resolution real smart meter data, which is released by a State Grid Corporation of China (SGCC) [29]. These data are time series, i.e., recorded at regular intervals of time. The input dimensions or features are 1032. The duration of collected data are three

years. It consists of electricity consumption data of 42,372 consumers. The released data also provide the ground truth according to which 9% of the total consumers are electricity thieves. This detail is given in Table 4. The daily electricity consumption pattern of the electricity thieves and honest consumers of over one month is given in Figure 3.

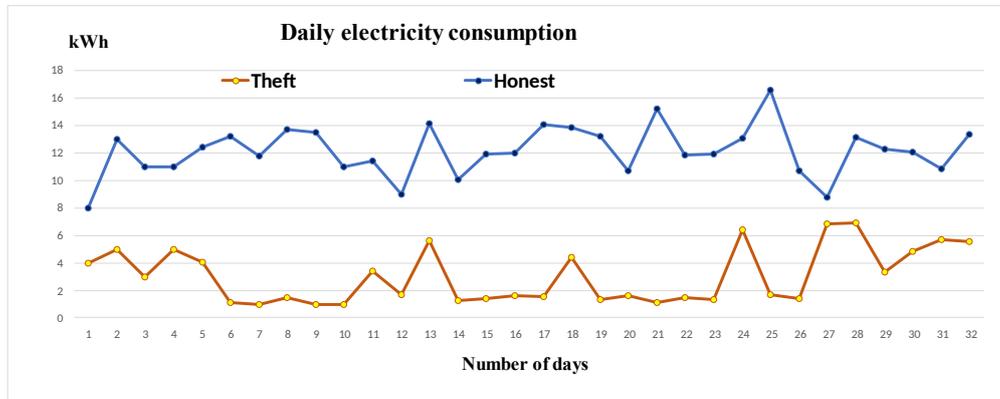


Figure 3. Electricity consumption pattern of thieves and honest consumers.

In the electricity consumption data, the honest consumers have different consumption patterns than the electricity thieves. The electricity thieves have irregular patterns of energy consumption and their amount of energy consumption is also low due to meter tempering. In contrast, the honest consumers have regular periodicity in their consumption pattern. The machine learning algorithms use the smart meter data to track the anomalous consumption pattern of consumers to identify the electricity thieves.

Table 4. Description of Data.

Description	Values
Duration of collected data	2014–2016
Data type	Time series
Dimension	1034
Samples	42,372
Resolution	High resolution real time smart meter data
Number of fraudulent consumers	3800
Number of honest consumers	38,530
Total consumers	42,372

2.1.2. Data Pre-Processing

In this paper, data pre-processing is performed to achieve better results in ETD. We exploit interpolation method [30] to recover the missing information using Equation (1):

$$f(x_i) = \begin{cases} \frac{(x_{i+1}+x_{i-1})}{2} & \text{if } x_i \in NaN, x_{i-1} \text{ and } x_{i+1} \notin NaN \\ 0 & \text{if } x_i \in NaN, x_{i-1} \text{ or } x_{i+1} \in NaN \\ x_i & \text{if } x_i \notin NaN. \end{cases} \quad (1)$$

where x_i is the attribute of the electricity consumption data and NaN represents the non-numeric value.

Afterwards, we use the three sigma rule to remove outliers from the raw data. These outliers show the peak electricity consumption that occurs during non-working days. We restore these values using Equation (2) according to the “Three sigma rule of thumb” [31],

$$f(x_i) = \begin{cases} avg(x) + 2std(x) & \text{if } x_i > avg(x) + 2std(x) \\ x_i & \text{else.} \end{cases} \quad (2)$$

In Equation (2), $std(x)$ is standard deviation and $avg(x)$ is the average value of x . This method is effective in handling the outliers.

Along with interpolation and the three sigma rule, we also used Min-Max scaling method to normalize the data between the range 0 and 1. It is important because neural networks show poor performance on inconsistent data [32]. The data normalization improves the training process of deep learning models by assigning a common scale to the data. The following equation is used to normalize the data [33]:

$$A' = \frac{A - Min(A)}{Max(A) - Min(A)}, \quad (3)$$

where A' is the normalized value. The performance of machine learning algorithms depends on quality of input data. Data pre-processing enhances the data quality and performance of these models.

2.1.3. Data Balancing

In this section, we deal with imbalance dataset. The dataset collected from SGCC has a larger number of normal electricity consumers than thieves. This data imbalance is a major problem in ETD, which needs to be resolved; otherwise, the classifier will be biased towards the majority class and can result in performance degradation [34]. Various Random Under Sampling (RUS) and Random Over Sampling (ROS) techniques are used in the literature to solve this problem.

In the RUS technique, the data samples from majority class are made equal to the minority class [35]. This technique reduces the size of dataset, which is computationally beneficial. However, this technique is not preferred. As it reduces the dataset, this gives the model less data to train on. In contrast, ROS replicates the minority class instances in order to balance the data. However, due to the replication of minority instances unintelligently, the model leads towards the overfitting problem. Another method used for the data balancing technique is SMOTE [36,37]. In this technique, the minority class instances are increased by finding the n -nearest neighbor samples in the same class, i.e., the theft class. The example of synthetic data generation is represented in Figure 4. It draws a line between the neighbor of the minority class instances and creates new points on the lines, which are the synthetic data samples. Synthetic generation of minority instances avoids the overfitting problem which occurs due to ROS technique; however, synthetic generation of NTL instances do not reflect real world theft cases. In addition, SMOTEBoost [38] creates synthetic examples from the minority class samples, which indirectly change the updating weights and compensate for the imbalanced distributions.

Motivated by SMOTE [37] and SMOTEBoost [38], which are helpful for handling the imbalanced data set, we use the Adasyn method [39] to balance the dataset. It is an enhanced version of SMOTE. With a minor modification, after creating n -nearest neighbors samples, it adds random values that are linearly correlated to the parent samples and have a little more variance. This modification generates more realistic data samples.

The Adasyn algorithm initially finds out the number of synthetic data samples g that need to be created to increase the minority class instances. It can be calculated using the following equation [39]:

$$g = (m_j - m_i)\beta, \quad (4)$$

where m_j and m_i are the numbers of majority and minority classes samples, respectively. $\beta \in [0,1]$ is a constraint used to set the balance level of minority class to the majority class. Next, we calculate the ratio r_i by finding K nearest neighbors, which is based on the Euclidean distance given in Equation (5) and mentioned in [39] as:

$$r_i = \delta i/k, \quad i = 1. \quad (5)$$

In the above equation, δi represents the synthetic samples and i represents the number of samples of the majority class in the k nearest neighbours; therefore, $r_i \in [0, 1]$. Finally, the number of synthetic data samples g_i are found by Equation (6) mentioned in [39] as:

$$g_i = r_i * g. \quad (6)$$

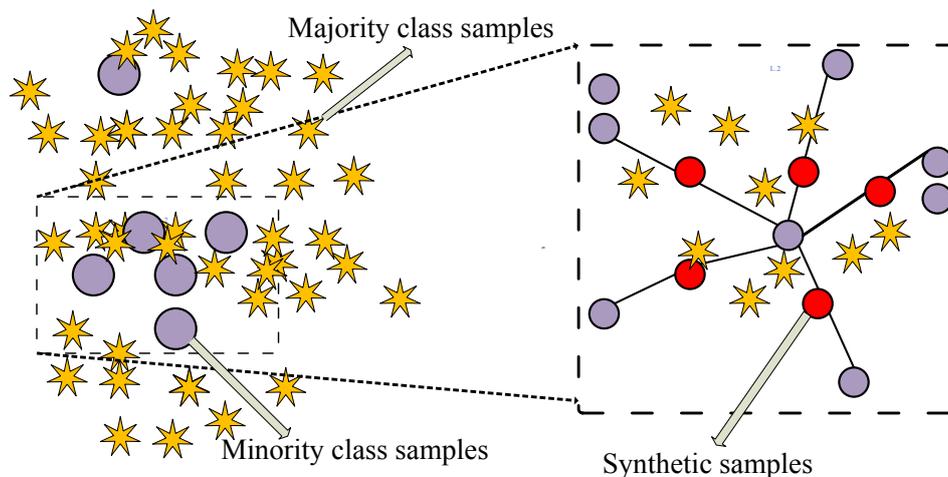


Figure 4. Generating of synthetic data through SMOTE.

The benefits of using Adasyn is two-fold; it improves the learning performance of the classifier to be more focused on theft cases that are harder to learn and prevents the model from being biased. The pseudo code of the Adasyn algorithm [40] is given in Algorithm 1.

Algorithm 1: Adasyn Algorithm

Input: Initial dataset X and desired balanced level β

Output: Synthetic dataset X_o

Initialize m_i as minority class samples

Initialize m_j as majority class samples

Synthesized total samples as $g = (m_j - m_i)\beta$

for each $X_i \in m_i$ **do**

 find the K nearest neighbors of m_i

$r_i = \delta i/k, \quad i = 1$

end for

for each $x_i \in m_i$ **do**

 select the synthetic samples

$g_i = r_i * g$

end for

return X_o

2.1.4. Feature Extraction Using VGG-16

VGG-16 is an enhanced version of CNN with 16 layers presented by the Visual Geometry Group [41]. It surpasses AlexNet by replacing large filters with small sized 3×3 filters [42]. It is

used for feature extraction and transfer learning [43,44]. In this paper, VGG-16 is used for feature extraction where the representation spaces constructed by all filters of a layer are visualized in more comprehensive ways. All activations of a layer are used to extract the relevant features through a deconvolution network.

The architecture of VGG-16 [44] is shown in Figure 5. It consists of the pooling layers and convolutional layers. The operations of all layers are summed up by three fully connected layers at the end. The softmax is used as the activation function in the final dense layer.

The multiple pooling layers used in the VGG-16 module are better at extracting the high level features from the input data. We can visualize what features each filter captures by learning the input image that maximizes the activation of that filter. The convolutional operation is performed by sliding the kernel over the entire input, which produces a feature map. The final output from the convolution layer is integrated after multiple operations of feature mapping by the kernel function, which is given in [19] as:

$$y = x \times F \rightarrow y[i] = \sum_{j=-\infty}^{+\infty} \times [i - j]F[j]. \tag{7}$$

In Equation (7), x is input and F is the filter, which is also called the kernel. The input image is initially random while the loss is calculated as the activation of a particular filter. Relu [19] is used as an activation function to introduce nonlinearity to the model:

$$Relu(x) = max(0, x). \tag{8}$$

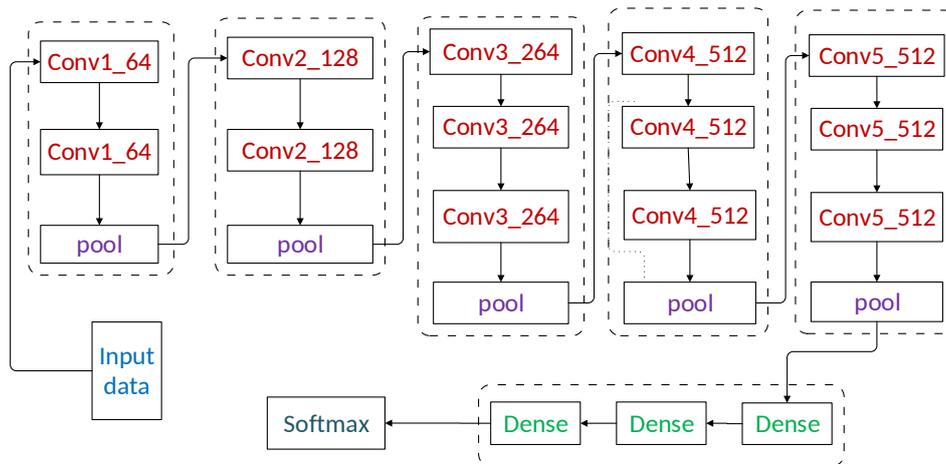


Figure 5. Architecture of VGG-16.

After the operations of pooling layers, three dense layers are used to visualize the important features. To avoid the overfitting problem, the dropout is set to 0.01 and the learning rate is 0.001. This method can be extended to the final dense layer having softmax as an activation function, which is defined in [19] as:

$$P(y = j|\varphi^{(i)}) = \frac{\varphi^{(i)}}{\sum_{k=0}^k \varphi_k^{(i)}}. \tag{9}$$

If the feature matrix and the weight matrix are denoted by X and W , then φ in the above equation is computed as:

$$\varphi = \sum_{i=1}^l W_i X_i = W^T X. \tag{10}$$

The hyper-parameters values of VGG-16 along with their description are given in Table 5. The hyper-parameters are batch size, learning rate, dropout rate, optimizer, and the number of epochs. These parameters play a key role in optimal performance of the VGG-16 module.

Table 5. Hyper-parameter values of VGG-16.

Hyper-Parameters	Values	Description
Batch size	130	It is training samples in each iteration
Leaning rate	0.001	It is a tuning parameter
Dropout	0.01	To avoid overfitting problem in neural networks.
Optimizer	Adam	It is adaptive learning rate.
Epochs	10	It is the number of iterations for training the algorithm

2.1.5. FA-XGBoost Based Classification

The XGBoost is one of the most popular machine learning methods [45]. On the Kaggle platform in 2015, the XGBoost as a classifier won 17 out of 29 competitions [45]. The extracted high level features given by VGG-16 become the inputs of the FA-XGBoost model. The FA-XGBoost library implements the gradient boosting decision tree algorithm. The ensemble model of XGBoost for classification is given in Figure 6. It shows that the XGBoost algorithm combines multiple weak models and makes a strong model to improve the final results. The final prediction is taken by voting of the majority of weak models.

There is a strong connection between hyper-parameters and outcome of a classifier [46]. Therefore, optimization is very important for accurate prediction. The hyper-parameters of XGBoost are learning rate and the number of estimators.

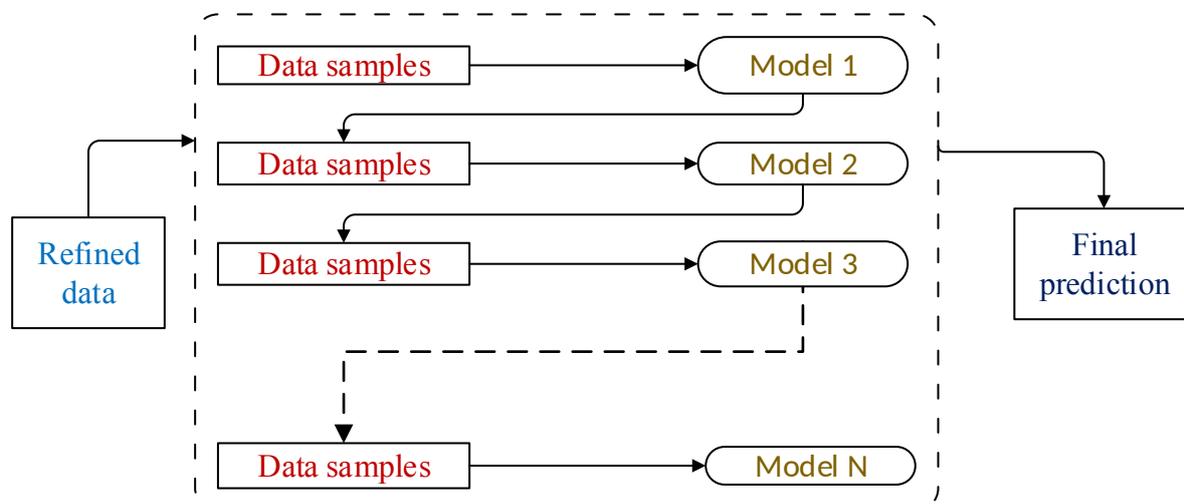


Figure 6. Ensemble model of XGBoost for classification.

The FA (developed by Yang [47]) is used in this paper to optimize the hyper-parameters of an XGBoost classifier. It is a nature inspired meta-heuristic algorithm based on flashing behavior of fireflies. The pseudo code of the FA-XGBoost [47] is given in Algorithm 2. The FA is based on three rules [48]:

1. Fireflies are uni-sexual in nature, so one firefly will be attracted to another regardless of whether the Firefly is male or female.

2. The attractiveness is proportional to light intensity of each firefly; thus, for any two flashing fireflies, the less bright firefly will be attracted by the brightest firefly. Attractiveness is calculated using Equation (11), which is mentioned in [49] as:

$$\beta(r) = \beta_0 e^{-\gamma r^2}. \quad (11)$$

In the above equation, $\beta(r)$ shows the attractiveness as a function of distance r , while β_0 represents attractiveness at zero distance. $e^{-\gamma r^2}$ is the value of rate of light absorption in the air.

3. As distance between fireflies increases, the attractiveness decreases. The distance r_{ij} between two fireflies i and j can be calculated using Euclidean distance as:

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2}, \quad (12)$$

where $x_{i,k}$ and $x_{j,k}$ are the t_{th} components of the position of fireflies i and j , respectively, while d is the number of dimensions. If no firefly is found brighter in the initialized population, then it moves in a random direction. The random movement towards the most brighter firefly is calculated using Equation (13), which is mentioned in [49] as:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} * (x_j^t - x_i^t) + \alpha * (rand - 1/2). \quad (13)$$

In the above equation, $rand$ represents the random number, t is the number of iterations, while α controls the size of random walk.

Algorithm 2: FA-XGBoost

- 1: Set the objective function by $y = (1,2,3 \dots n)$
 - 2: Initialize the population of Fireflies by y_i ($i = 1,2,3 \dots n$)
 - 3: Define γ as the rate of light absorption in the air
 - 4: Define I as the light intensity of a firefly
 - 5: Maximum number of iteration is m and t is current iteration
 - 6: **while** ($t < m$)
 - 7: **for** $i = 1$:
 - 8: **for** $j = 1$:
 - 9: **if** ($I_i > I_j$) **then**
 - 10: Move Firefly i towards j
 - 11: **end if** ;
 - 12: Attractiveness varies with distance r as given in Equation (8)
 - 13: Adjust the light intensity I to find new solutions
 - 14: Choose the best solution by random fly
 - 15: **end for** j
 - 16: **end for** i
 - 17: Rank the Fireflies on the basis of minimum cost function
 - 18: Choose the current best solution
 - 19: **end while**
 - 20: Return the best values of performance metrics
-

The XGBoost model for classification is given in Figure 6. It is based on ensemble learning in which several weak classifiers are combined to make a strong classifier. On each iteration, the classification rate of each learner is computed. The predicted value y_i after k iterations is computed using Equation (14):

$$\bar{y}_i = \sum_{k=1}^k [f_k(x_i)], \quad (14)$$

where $f_k(x_i)$ is the input function. The loss function $\mathcal{L} \phi$ is calculated by taking the difference between the actual y_i and predicted result \bar{y}_i as given is Equation (15):

$$\mathcal{L}(\phi) = \sum_{i=1}^l [l(\bar{y}_i, y_i)]. \quad (15)$$

The objective of XGBoost is to minimize the loss function given Equations (16)–(18), which is computed by taking the summation of loss l of the multiple weak learners:

$$\min \mathcal{L}(f_t) = \min(f_t) \sum_{i=1}^l [l(\bar{y}_i, y_i)], \quad (16)$$

$$= \min(f_t) \sum_{i=1}^l [l(y_i, \sum_{i=1}^l [l(f_k(x_i))])], \quad (17)$$

$$= \min(f_t) \sum_{i=1}^l [l(y_i^{t-1} \bar{y}_i, f_t(x_i))]. \quad (18)$$

The instances of electricity theft that are misclassified by the learner are given more weight in the next iteration. The weights are adjusted by using the penalizes function $\Omega(f)$, which can be calculated using the following equation:

$$\Omega(f) = \gamma T + 1/2\lambda \|w\|^2. \quad (19)$$

In Equation (19), γ and λ are the hyper-parameters, T is the number of tree node, and W is the vector of the nodes. When the penalized function is added to the loss, it minimizes the objective function and helps to smooth the final learnt weight:

$$\mathcal{L}(\phi) = \min(f_t) \sum_{i=1}^l [l(\bar{y}_i, y_i)] + \sum_{i=1}^l \Omega f_k. \quad (20)$$

The final classification is performed by taking the mean of individual models. The prediction of each individual decision tree is weak and prone to overfitting. However, combining several decision trees in an ensemble method gives better results. For comparative analysis, various performance metrics are used, i.e., F1-score, precision, recall, and ROC curve to validate the effectiveness of our proposed model. They are discussed in detail in the simulation section.

3. Experiments and Results

In this section, the experimental results are discussed in detail.

3.1. Loss Function

For accurate prediction, the proposed model aims to reduce the loss function. The widely used logarithmic loss function is cross entropy. As we are doing binary classification, the loss function we are using is a binary cross entropy. It is calculated using the following equation [50]:

$$f(x) = 1/N \sum_{i=1}^N -(y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))). \quad (21)$$

In Equation (21), N is the total number of consumers samples, $p(y_i)$ is the probability of electricity theft, and y_i is the ground truth label.

3.2. Model Evaluation Metrics

The concern of ETD in supervised learning is a class imbalance problem. In this problem, the number of honest customers varies remarkably from these fraudulent ones. Therefore, for evaluation, a simple accuracy measure is not reliable. In this paper, various performance metrics are considered. These evaluation metrics' values are determined from confusion matrix. The confusion matrix gives information about the following results:

- True positive (TP), the dishonest consumers accurately predicted as dishonest.
- True Negative (TN), the honest consumers accurately predicted as honest.
- False Positive (FP), the honest consumers predicted as thieves.
- False Negative (FN), the dishonest consumers predicted as honest consumers.

In this paper, we use precision, recall, F1-score, ROC-AUC, and MCC for evaluation of our system model. Precision is referred to as True Negative Rate (TNR); it shows the actual number of honest customers that are correctly identified by the classifier. It is formulated in [19] using Equation (22). Recall is referred to as True Positive Rate (TPR); it shows the actual number of positives that are correctly identified by classifier. It is formulated in [19] using Equation (23). Both precision and recall are not enough to show real assessment of a classifier. It is better to maximize precision and recall, which gives F1-score. It is a useful measure for binary classification problems where the distribution of classes is imbalanced. It is calculated by the weighed harmonic mean of precision and recall [19], which is given in Equation (24). Another suitable metric for ETD is the ROC-AUC. It shows a graphical representation of a model to evaluate its detection performance. The classifier having ROC-AUC close to 1 has better capability to separate two classes. However, ROC-AUC only summarizes the trade-off between the TPR and FPR of the model. AUC score is calculated by using Equation (25), mentioned in [30]. In Equation (25), Rank shows the number of samples, M is the number of positive class samples, and N is the number of negative class samples. Moreover, the PR-AUC are appropriate for imbalanced datasets. Its graphical representation is obtained by plotting the recall against the precision. The value of curve is in the range between 0 and 1. The classifier having ROC-AUC value close to 1 is considered a good classifier. In all performance matrices, MCC produces a high score only if the prediction obtained good results in all of the four confusion matrix values, i.e., TP , TN , FP , and FN . MCC score ranges between -1 to 1 , whereas, close to 1 shows the accurate classification, 0 shows no class separation capability, and -1 shows the incorrect classification by model. It is calculated using Equation (26) mentioned in [19]. Accuracy is the number of correctly predicted data points out of all the data points. It is a widely used metric for classification problems in the data science community [51]. However, it is not considered as a reliable metric where the distribution of labels is imbalanced. It can be calculated by using Equation (27):

$$Precision = \frac{TP}{TP + FP}. \quad (22)$$

$$Recall = \frac{TP}{TP + FN}. \quad (23)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (24)$$

$$AUC = \frac{\sum Rank_{i \in positive class} - \frac{M(1+M)}{2}}{M * N}. \quad (25)$$

$$MCC = \frac{TP * TN - FR * FN}{\sqrt{((TP + FP)(TP + FN)(TN + FP)(TN + FN))}}. \quad (26)$$

$$Accuracy = \frac{TP * TN}{TP + TN + FP + FN}. \quad (27)$$

3.3. Benchmark Models and Their Configuration

In this section, we describe the conventional models, which are widely used as classifiers for ETD. The range of hyper-parameter values is defined, and we select optimal values for each base model.

3.3.1. SVM Model

It is a popular classifier and widely used for ETD [18]. The hyper-parameters of SVM are γ and regularization parameter C , which are important in selecting an optimal hyperplane. The values of these parameters are given in Table 6. The optimal values are selected from the given range of values.

Table 6. Hyper-parameter values of SVM.

Hyper-Parameters	Range of Values	Selected Value
γ	1,3,5	3
C	0.001, 0.01,	0.01

3.3.2. LR Model

It is a supervised learning algorithm, which is used as benchmark model in this paper. Its hyper-parameters along with their range and selected values are given in Table 7. During implementation, we choose optimal values for accurate classification.

Table 7. Hyper-parameter values of LR.

Hyper-Parameters	Range of Values	Selected Value
R	0.001, 0.01, 0.1	0.001
C	l1 norm, l2 norm	l2 norm

3.3.3. RUSBoost Model

It is widely used for classification problems where the distribution of labels is imbalanced. The hyper-parameters of RUSBoost are learning rate and number of estimators. The best values are selected from the range of different values as given in Table 8.

Table 8. Hyper-parameter values of RUSBoost.

Hyper-Parameters	Range of Values	Selected Value
Learning rate	0.2, 0.5, 1	1
Estimator	150, 200, 300	200

3.3.4. CNN Model

Along with conventional machine learning algorithms, we also used CNN as a deep learning model for comparison. It is a feed-forward neural network and is mostly used for complex classification problems. We choose best values of CNN during model validation, which are given in Table 9.

Table 9. Hyper-parameter values of CNN.

Hyper-Parameters	Range of Values	Selected Value
Epochs	10,15,30	10
Batch size	50, 80,130	50
Dropout	0.01, 0.1, 0.2	0.2

3.4. Proposed Model Results

In this section, we present the performance of our proposed model on raw data and transformed data. Initially, the missing values in the data set are filled with interpolation and three sigma rule methods. In addition, the data are normalized using the Min-Max scaling method. Figure 7a,b shows the unbalanced and balanced distribution of labels, respectively. The thieves are represented by '0', while '1' shows honest customers. The x -axis represents observation values for the first sample, and the y -axis represents the observation values for the second sample. Each point on the plot represents a single observation.

The Adasyn algorithm is used to address the class imbalance problem. This algorithm intelligently balances the number of instances of electricity thieves and honest consumers. The distribution of two classes after applying the Adasyn algorithm is also shown in Figure 7b. The minority class instances are increased, and it shows the equal distribution of the two labels in the data.

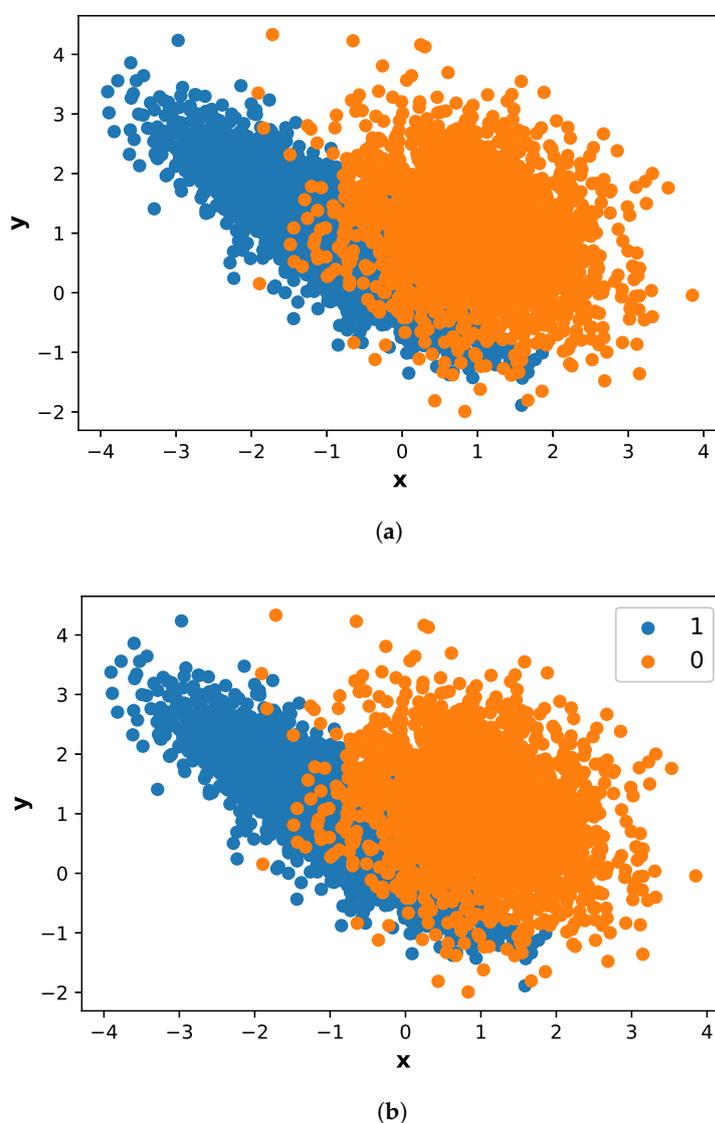


Figure 7. (a) visualization of unbalanced data; (b) visualization of balanced data.

Due to imbalanced data, the classifiers get biased and result in high FPR. In order to show the effectiveness of balanced data, we make a comparison that is shown in Table 10. Before applying Adasyn, the model could not classify effectively as evident from the scores mentioned in Table 10 and Figure 8. On imbalance data, the classifier achieves 60% precision, 62.1% recall, 59.01% F1-score,

and the 63.2% ROC curve. The SMOTE improves the performance of the FA-XGBoost classifier. It achieves 79.1% precision, 80% recall, 78.7% F1-score, and 78% ROC curve. Adasyn improves the ability of the model by using synthetic data intelligently. The results of Adasyn are far better as compared to the results of the unbalanced data and SMOTE. On the imbalance dataset, the classifier becomes biased by considering the real electricity thieves as honest consumers. The Adasyn method improved the performance of the FA-XGBoost classifier. It achieves 93% precision, 97% recall, 93.7% F1-score, and the 95.9% ROC curve.

Table 10. Model performance before and after Adasyn.

Performance Metrics	Imbalanced Data	SMOTE	Adasyn
Precision	60	79.1	93
Recall	62.1	80	97
F1-score	59.01	78.7	93.7
ROC-AUC	63.2	78	95.9

Our goal in ETD is to maximize the *TP* and *TN* and reduce the *FP* and *FN*. Table 11 shows that our proposed model has achieved good values of the confusion matrix. The high *TP* and *TN* values show that our model has truly identified the electricity thieves and honest consumers, respectively.

Table 11. Confusion matrix values of the ETD model. *TN*: True Negative; *FP*: False Positive; *FN*: False Negative; *TP*: True Positive.

Confusion Matrix	Predicted No	Predicted Yes
Actual No	<i>TN</i> = 9306	<i>FP</i> = 1296
Actual Yes	<i>FN</i> = 948	<i>TP</i> = 8996

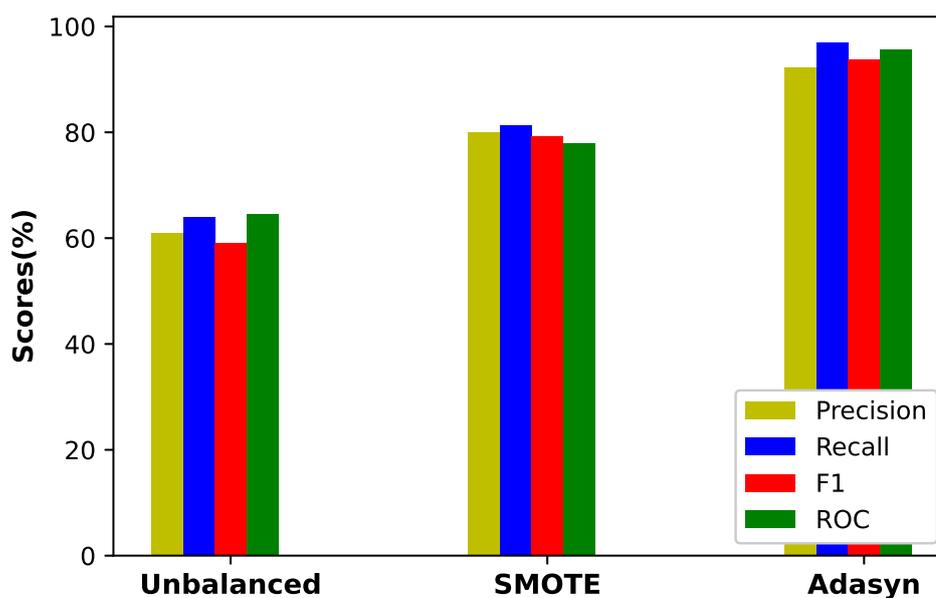


Figure 8. Performance comparison of balanced and unbalanced datasets.

In this paper, FA-XGBoost is used as a classifier. Its performance is primarily dependent on the selection of hyper-parameter values. Initially, we randomly apply the XGBoost without tuning its hyper-parameter values. Still, it achieves better performance than the state-of-the-art models, i.e., 86.5% ROC-AUC. To enhance the classification performance, we utilize the Firefly algorithm to choose the optimal hyper-parameter values of XGBoost. The results of ROC-AUC of FA-XGBoost in Figure 9 shows the better result, i.e., 95.6% ROC-AUC.

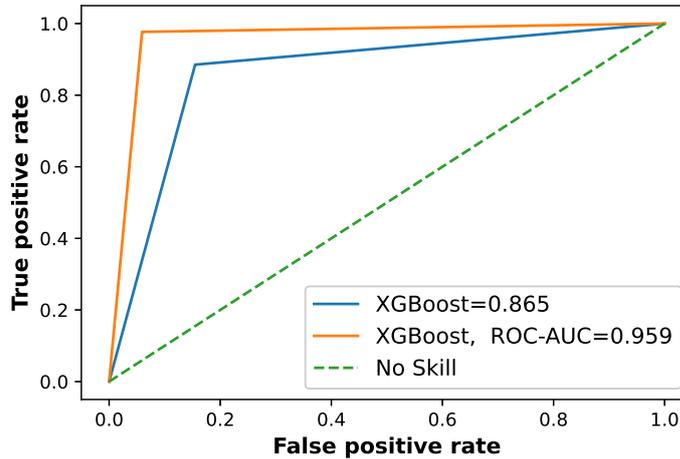


Figure 9. Performance comparison of parameter tuning.

3.5. Convergence Analysis

As expected, using VGG-16 for feature extraction improves the performance for ETD. Figure 10 shows the accuracy and loss of VGG-16 module. When we choose a smaller epoch value to optimize the training procedure, it is enough to let our model to learn from high dimensional data. However, it causes overfitting when we choose a larger epoch value. As the number of epochs increases up to four, the training and testing losses decrease and the accuracy increases significantly. It shows the better prediction capability of the model. The mapping of addressed problems to the validation results is presented in Table 12. There is no direct validation for pre-processing methods. The class imbalance problem is efficiently solved through the Adasyn method that is validated in Figure 8. The overfitting problem is solved by achieving a higher generalized performance through the VGG-16 module. In addition, the Firefly based XGBoost classifier enhanced the classification accuracy as shown in Figure 9.

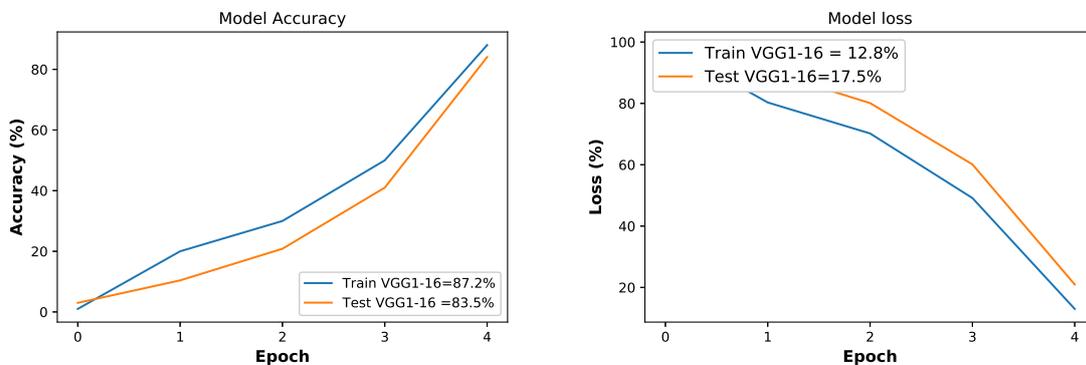
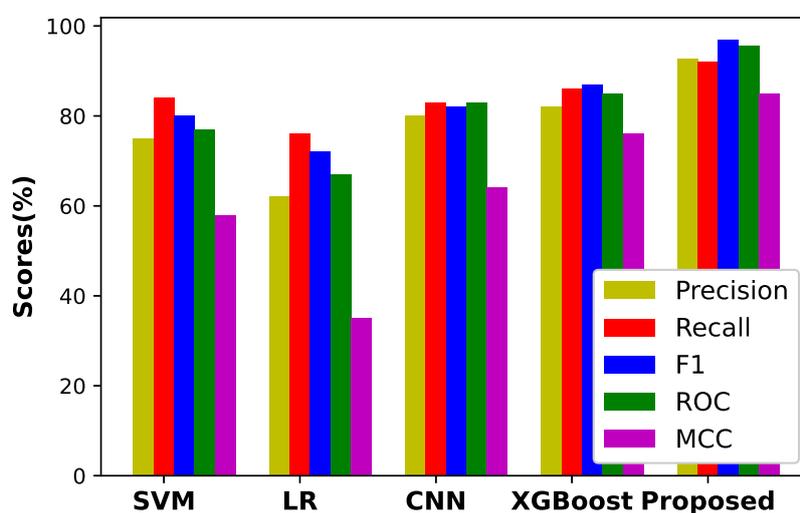


Figure 10. Accuracy and loss of VGG-16.

Table 12. Mapping of problems addressed and validation results.

Limitation Number	Limitation Identified	Solution Number	Validation Results
L.1	Missing values and outliers	S.1	No direct validation
L.2	Imbalanced data	S.2	Adasyn algorithm effectively handles imbalance data as shown in Figure 8
L.3	Overfitting	S.3	Figure 10 shows a generalized performance of our proposed model
L.4	Poor classification	S.4	Firefly based XGBoost classifier achieved excellent results in terms of all performance metrics as mentioned in Figure 9
L.5	No reliable Evaluation	S.5	Figure 11 shows the performance of our proposed model in terms of several performance metrics

**Figure 11.** Performance metrics comparison with benchmark schemes.

3.6. Comparison with Benchmark Models

In order to show the comparison of our proposed model with benchmark schemes, we trained LR, SVM, CNN, and RUSBoost using the same dataset. These are basic models for classification problems. The configuration of these models is discussed in Section 3.4.

Figure 12 shows ROC-AUC of SVM, CNN, LSTM-RUSBoost, and LR models. We obtained the results by using the same dataset and setting optimal hyper-parameters of these models. It can be seen that RUSBoost outperforms the benchmark schemes by achieving 86.5% ROC-AUC. It balances the data by a random under sampling method. It performs classification through the adaptive boosting technique. This technique is better for unbalanced binary classification problems.

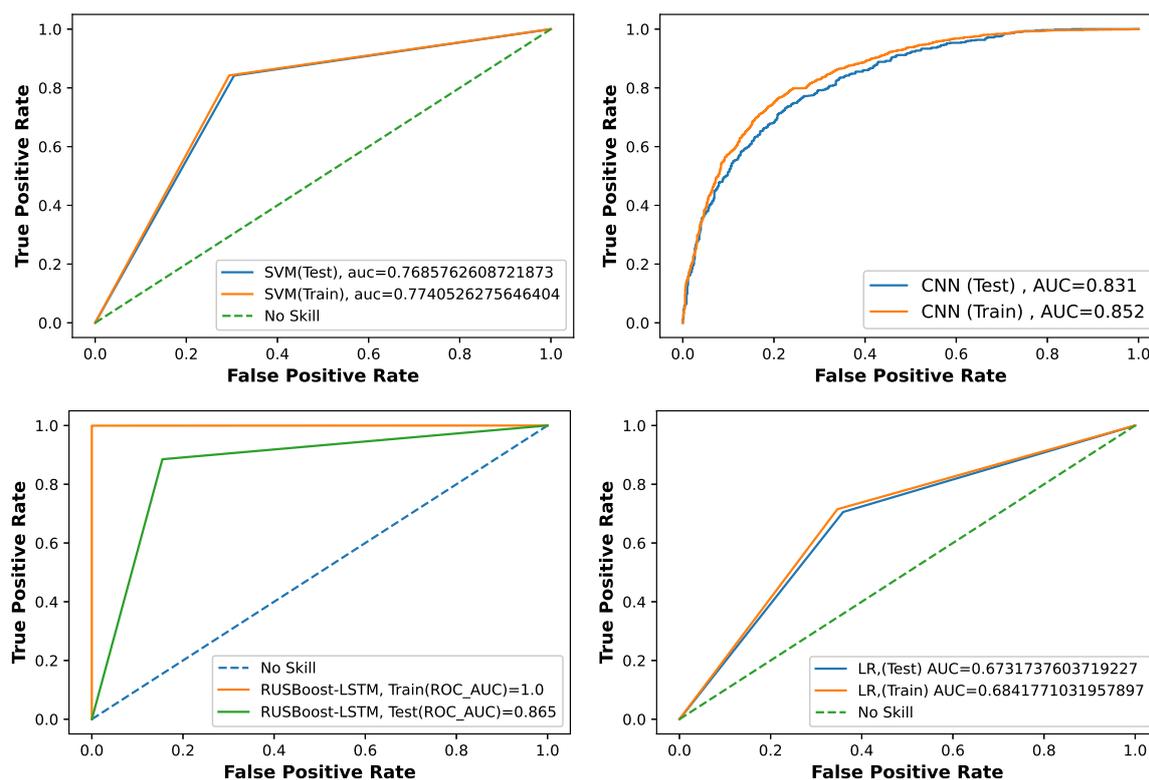


Figure 12. ROC-AUC of benchmark models.

However, LR performs worst among the classifiers, securing just 67.3 % ROC-AUC. It is due to the fact that LR is based on the concept of probability and uses the principle of neural networks, which do not capture the long-term dependencies from the large time series data. Moreover, it becomes biased during the identification of real electricity theft cases due to the training on the majority class samples. Hence, LR can not perform accurate classification on the large imbalanced dataset.

The performance of SVM is also not satisfactory, securing 76.8% ROC-AUC. It classifies the data by creating a hyperplane. However, in a complex binary classification problem, it becomes difficult for SVM to set optimal values for creating a hyperplane. Hence, this method is also not suitable for ETD. In contrast, the CNN slightly performs better than SVM by securing 83.1% ROC-AUC. The CNN is a learning deep model having multiple stacks of hidden layers that extract the hidden patterns from the electricity consumption data and identify the real electricity thieves. However, it has over fitting issues due to the dense layers. It fails to achieve a generalized performance.

Figure 11 presents the performance comparison of our proposed scheme with benchmark models. It is worth noting that our proposed model outperforms the other models in terms of precision, recall, ROC-AUC, and F1-score. The Firefly based XGBoost achieves 95.9% ROC-AUC, 92.6% precision, 97% recall, and 93.7% F1-score.

The ROC-AUC and PR-AUC of benchmark schemes and our proposed model is shown in Figure 13. It can be seen that deep learning models like CNN perform better for the classification of high dimensional data. The CNN achieves 83.1% ROC-AUC on the test dataset. These models automatically extract the features from the data while traditional machine learning algorithms require the separate techniques for refining the data. They capture long-term dependencies and improve the performance as the dataset increases. Moreover, the traditional machine learning algorithm like SVM and LR are not efficient for the classification using a larger dataset. The SVM and LR get 76% ROC-AUC and 67.3% ROC-AUC, respectively.

In this paper, we also evaluate our proposed model on the PR-AUC. Our proposed model also covers more area under the PR-AUC than the benchmark models as shown in Figure 13. The results of

ROC-AUC and PR-AUC show that our proposed model is superior to other classifiers. The overall summary of the benchmark model and the proposed model is presented in Table 13. As it can be observed, FA-XGBoost outperforms the rest of the classifiers in terms of all the performance metrics. The high values of precision, recall, F1, and ROC-AUC show that our model has truly identified the number of honest consumers and electricity thieves.

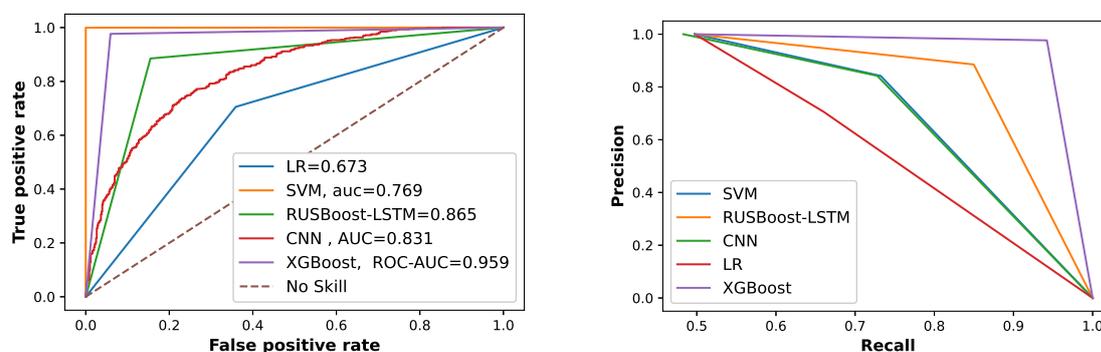


Figure 13. ROC-AUC and PR-AUC comparison.

In this paper, our focus was more on accurate identification of electricity thieves. However, our proposed model has high execution time. It takes 25 min to run the model.

Table 13. Summary of results.

Models	Accuracy	Precision	Recall	F1-Score	ROC	MCC
CNN	0.812	0.805	0.862	0.845	0.813	61.5
SVM	0.772	0.765	0.883	0.819	0.769	56.3
LR	0.676	0.645	0.772	0.701	0.673	35.6
RUSBoost	0.869	0.85	0.896	0.871	0.865	77.8
Proposed Model	0.95	0.930	0.9700	0.937	0.959	85.6

4. Conclusions and Future Work

In this paper, the proposed methodology is implemented for ETD using the real smart meter data. The different limitations in literature are addressed in this work. The conclusions were drawn and are summarised as follows.

Initially, the real smart meter data, which is collected from SGCC, have a number of missing values and outliers. For this reason, we performed a comprehensive data pre-processing which consists of interpolation, the three sigma rule, and normalization methods. In addition, the dataset has a small number of instances for electricity thieves, which makes the classification model biased due to its training on majority honest instances. We employed the Adaysn algorithm to address this problem. This technique has improved the performance of the FA-XGboost classifier, which has achieved F1-score, precision, and recall of 93.7%, 92.6%, and 97%, respectively. Afterwards, the model has overfitting issues due to training the model on a large time series data, a VGG-16 module is introduced in ETD, which extracts relevant features from the data. It achieved a higher generalized performance by securing accuracy of 87.2% and 83.5% on training and testing data, respectively. Finally, the XGBoost method is applied to classify data into honest and dishonest consumers. To enhance the performance of XGBoost method, an FA is used for parameters' optimization. This method improved the performance of XGBoost and achieved 95.9% ROC-AUC and outperforming the benchmarks: SVM, LR, and CNN. However, as the dataset increases, the execution time of our proposed model

also increases. In the future, we will improve its performance by reducing the delay in detecting the electricity theft.

Author Contributions: Z.A.K. and M.A. proposed and implemented the main idea. N.J. and M.N.S. performed the mathematical modeling and wrote the simulation section. M.S. and J.-G.C. organized and refined the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Basic Science Research Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Education under Grant 2018R1D1A1B07048948.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Gul, H.; Javaid, N.; Ullah, I.; Qamar, A.M.; Afzal, M.K.; Joshi, G.P. Detection of Non-Technical Losses using SOSTLink and Bidirectional Gated Recurrent Unit to Secure Smart Meters. *Appl. Sci.* **2020**, *10*, 3151. [[CrossRef](#)]
- Adil, M.; Javaid, N.; Qasim, U.; Ullah, I.; Shafiq, M.; Choi, J.-G. LSTM and Bat-Based RUSBoost Approach for Electricity Theft Detection. *Appl. Sci.* **2020**, *10*, 1–21.
- Mujeeb, S.; Javaid, N. ESAENARX and DE-RELM: Novel schemes for big data predictive analytics of electricity load and price. *Sustain. Cities Soc.* **2019**, *51*, 101642. [[CrossRef](#)]
- Nazari-Heris, M.; Mirzaei, M.A.; Mohammadi-Ivatloo, B.; Marzb, M.; Asadi, S. Economic-environmental effect of power to gas technology in coupled electricity and gas systems with price-responsive shiftable loads. *J. Clean. Prod.* **2020**, *244*, 118769. [[CrossRef](#)]
- Marzb, M.; Azarinejadian, F.; Savaghebi, M.; Pouresmaeil, E.; Guerrero, J.M.; Lightbody, G. Smart transactive energy framework in grid-connected multiple home microgrids under independent and coalition operations. *Renew. Energy* **2018**, *126*, 95–106.
- Jadidbonab, M.; Mohammadi-Ivatloo, B.; Marzb, M.; Siano, P. Short-term Self-Scheduling of Virtual Energy Hub Plant within Thermal Energy Market. *IEEE Trans. Ind. Electron.* **2020**, accepted. [[CrossRef](#)]
- Gholinejad, H.R.; Loni, A.; Adabi, J.; Marzb, M. A hierarchical energy management system for multiple home energy hubs in neighborhood grids. *J. Build. Eng.* **2020**, *28*, 101028. [[CrossRef](#)]
- Mirzaei, M.A.; Sadeghi-Yazdankhah, A.; Mohammadi-Ivatloo, B.; Marzb, M.; Shafie-khah, M.; Catalão, J.P. Integration of emerging resources in IGD-based robust scheduling of combined power and natural gas systems considering flexible ramping products. *Energy* **2019**, *189*, 116195. [[CrossRef](#)]
- Biswas, P.P.; Cai, H.; Zhou, B.; Chen, B.; Mashima, D.; Zheng, V.W. Electricity Theft Pinpointing through Correlation Analysis of Master and Individual Meter Readings. *IEEE Trans. Smart Grid* **2019**, *11*, 3031–3042. [[CrossRef](#)]
- Lydia, M.; Kumar, G.E.P.; Levron, Y. Detection of Electricity Theft based on Compressed Sensing. In Proceedings of the 2019 5th International Conference on Advanced Computing and Communication Systems (ICACCS) IEEE, Coimbatore, India, 15–16 March 2019; pp. 995–1000.
- Razavi, R.; Gharipour, A.; Fleury, M.; Akpan, I.J. A practical feature-engineering framework for electricity theft detection in smart grids. *Appl. Energy* **2019**, *238*, 481–494. [[CrossRef](#)]
- Depuru, S.S.S.R.; Wang, L.; Devabhaktuni, V. Support vector machine based data classification for detection of electricity theft. In Proceedings of the 2011 IEEE/PES Power Systems Conference and Exposition, Phoenix, AZ, USA, 20–23 March 2011; pp. 1–8.
- Saeed, M.S.; Mustafa, M.W.; Sheikh, U.U.; Jumani, T.A.; Mirjat, N.H. Ensemble Bagged Tree Based Classification for Reducing Non-Technical Losses in Multan Electric Power Company of Pakistan. *Electronics* **2019**, *8*, 860. [[CrossRef](#)]
- Razavi, R.; Fleury, M. Socio-economic predictors of electricity theft in developing countries: An Indian case study. *Energy Sustain. Dev.* **2019**, *49*, 1–10. [[CrossRef](#)]
- McDaniel, P.; McLaughlin, S. Security and privacy challenges in the smart grid. *IEEE Secur. Priv.* **2009**, *7*, 75–77. [[CrossRef](#)]
- Buzau, M.M.; Tejedor-Aguilera, J.; Cruz-Romero, P.; Gomez-Exposito, A. Hybrid deep neural networks for detection of non-technical losses in electricity smart meters. *IEEE Trans. Power Syst.* **2019**, *35*, 1254–1263. [[CrossRef](#)]

17. Jamil, A.; Alghamdi, T.A.; Khan, Z.A.; Javaid, S.; Haseeb, A.; Wadud, Z.; Javaid, N. An Innovative Home Energy Management Model with Coordination among Appliances using Game Theory. *Sustainability* **2019**, *11*, 6287. [CrossRef]
18. Buzau, M.M.; Tejedor-Aguilera, J.; Cruz-Romero, P.; Gómez-Expósito, A. Detection of non-technical losses using smart meter data and supervised learning. *IEEE Trans. Smart Grid* **2018**, *10*, 2661–2670. [CrossRef]
19. Hasan, M.; Toma, R.N.; Nahid, A.A.; Islam, M.M.; Kim, J.M. Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach. *Energies* **2019**, *12*, 3310. [CrossRef]
20. Avila, N.F.; Figueroa, G.; Chu, C.C. NTL detection in electric distribution systems using the maximal overlap discrete wavelet-packet transform and random under sampling boosting. *IEEE Trans. Power Syst.* **2018**, *33*, 7171–7180.
21. Ramos, C.C.; Rodrigues, D.; de Souza, A.N.; Papa, J.P. On the study of commercial losses in Brazil: a binary black hole algorithm for theft characterization. *IEEE Trans. Smart Grid* **2016**, *9*, 676–683. [CrossRef]
22. Zheng, K.; Chen, Q.; Wang, Y.; Kang, C.; Xia, Q. A novel combined data-driven approach for electricity theft detection. *IEEE Trans. Ind. Inform.* **2019**, *15*, 1809–1819. [CrossRef]
23. Ding, N.; Ma, H.; Gao, H.; Ma, Y.; Tan, G. Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model. *Comput. Electr. Eng.* **2019**, *70*, 106458. [CrossRef]
24. Li, S.; Han, Y.; Yao, X.; Yingchen, S.; Wang, J.; Zhao, Q. Electricity Theft Detection in Power Grids with Deep Learning and Random Forests. *J. Electr. Comput. Eng.* **2019**, *2019*, 1–12. [CrossRef]
25. Punmiya, R.; Choe, S. Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. *IEEE Trans. Smart Grid* **2019**, *10*, 2326–2329. [CrossRef]
26. Amin, S.; Schwartz, G.A.; Cardenas, A.A.; Sastry, S.S. Gametheoretic models of electricity theft detection in smart utility networks: Providing new capabilities with advanced metering infrastructure. *IEEE Control. Syst. Mag.* **2015**, *35*, 66–81.
27. Leite, J.B.; Mantovani, J.R.S. Detecting and locating non-technical losses in modern distribution networks. *IEEE Trans. Smart Grid* **2016**, *9*, 1023–1032. [CrossRef]
28. Wang, S.; Chen, H. A novel deep learning method for the classification of power quality disturbances using deep convolutional neural network. *Appl. Energy* **2019**, *235*, 1126–1140. [CrossRef]
29. State Grid Corporation of China. Available online: <https://www.sgcc.com.cn> (accessed on 22 February 2020).
30. Zheng, Z.; Yang, Y.; Niu, X.; Dai, H.N.; Zhou, Y. Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. *IEEE Trans. Ind. Informat.* **2017**, *14*, 1606–1615. [CrossRef]
31. Chola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *Acm Comput. Surv. (Csur)* **2009**, *41*, 1–58.
32. Nam, H.; Kim, H.E. Batch-instance normalization for adaptively style-invariant neural networks. In *Advances in Neural Information Processing Systems*; The MIT Press: Cambridge, MA, USA, 2018; pp. 2558–2567.
33. Pandey, A.; Jain, A. Comparative analysis of KNN algorithm using various normalization techniques. *Int. J. Comput. Netw. Inf. Secur.* **2017**, *9*, 36–42. [CrossRef]
34. Figueroa, G.; Chen, Y.S.; Avila, N.; Chu, C.C. Improved practices in machine learning algorithm for NTL detection with imbalanced data. In Proceedings of the 2017 IEEE Power Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017; pp. 1–5.
35. Hasanin, T.; Khoshgoftaar, T. The effects of random under sampling with simulated class imbalance for big data. In Proceedings of the 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, USA, 6–9 July 2018; pp. 70–79.
36. Qin, H.; Zhou, H.; Cao, J. Imbalanced Learning Algorithm based Intelligent Abnormal Electricity Consumption Detection. *Neurocomputing* **2020**, *402*, 112–123.
37. Qu, Z.; Li, H.; Wang, Y.; Zhang, J.; Abu-Siada, A.; Yao, Y. Detection of Electricity Theft Behavior Based on Improved Synthetic Minority Oversampling Technique and Random Forest Classifier. *Energies* **2020**, *13*, 1–20. [CrossRef]
38. Pelayo, L.; Dick, S. Synthetic minority oversampling for function approximation problems. *Int. J. Intell. Syst.* **2019**, *34*, 2741–2768. [CrossRef]
39. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE world Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008.
40. Xiang, Y. *Polarity Classification of Imbalanced Microblog Texts*; AIST: Tsukuba, Ibaraki, Japan, 2019; pp. 1–61.

41. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
42. Yu, W.; Yang, K.; Bai, Y.; Xiao, T.; Yao, H.; Rui, Y. Visualizing and comparing AlexNet and VGG using deconvolutional layers. In Proceedings of the 33rd International Conference on Machine Learning, New York City, NY, USA, 19–24 June 2016; pp. 1–7.
43. Dixon, J.; Rahman, M. Modality Detection and Classification of Biomedical Images with Deep Transfer Learning and Feature Extraction. In Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCVC) The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), Las Vegas, NV, USA, 29 July–1 August 2019; pp. 55–58.
44. Cibuk, M.; Budak, U.; Guo, Y.; Ince, M.C.; Sengur, A. Efficient deep features selections and classification for flower species recognition. *Measurement* **2019**, *137*, 7–13. [[CrossRef](#)]
45. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
46. Zahid, M.; Ahmed, F.; Javaid, N.; Abid Abbasi, R.; Zainab Kazmi, H.S.; Javaid, A.; Bilal, M.; Akbar, M.; Ilahi, M. Electricity Price and Load Forecasting using Enhanced Convolutional Neural Network and Enhanced Support Vector Regression in Smart Grids. *Electronics* **2019**, *8*, 122. [[CrossRef](#)]
47. Yang, X.-S. Firefly Algorithm, Stochastic Test Functions and Design Optimization. *Int. Bio-Inspired Comput.* **2010**, *2*, 78–84. [[CrossRef](#)]
48. Yang, X.S. Chaos-enhanced firefly algorithm with automatic parameter tuning. In *Recent Algorithms and Applications in Swarm Intelligence Research*; Information Science Reference (IGI Global): Hershey, PA, USA, 2013; pp. 125–136.
49. Chen, K.; Zhou, Y.; Zhang, Z.; Dai, M.; Chao, Y.; Shi, J. Multilevel image segmentation based on an improved firefly algorithm. *Math. Probl. Eng.* **2016**, *2016*, 1–12. [[CrossRef](#)]
50. Janocha, K.; Czarnecki, W.M. On loss functions for deep neural networks in classification. *arXiv* **2017**, arXiv:1702.05659.
51. Zhu, W.; Zeng, N.; Wang, N. Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *Nesug Proc. Health Care Life Sci. Balt. Md.* **2010**, *19*, 67.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).