Electrical Engineering (2024) 106:5987–6001
https://doi.org/10.1007/s00202-024-02342-7

ORIGINAL PAPER

Electricity theft detection in IoT-based smart grids using a parameter-tuned bidirectional LSTM with pre-trained feature learning mechanism

Mahendran Krishnamoorthy[1] · Johny Renoald Albert[2]

Received: 22 July 2023 / Accepted: 29 February 2024 / Published online: 29 March 2024

Abstract
The most significant issue today is electricity theft (ET) which causes much loss to electricity boards. The development of smart grids (SGs) is crucial for ET detection (ETD) because these systems produce enormous amounts of data, including information on customer consumption, which can be used to identify ET using machine learning and deep learning (DL) techniques. However, the existing models majorly suffers with lower prediction accuracy because of over-fitting and dataset imbalancing issues. Therefore, to overcome these shortcomings, this paper proposes a novel DL approach for ETD in the Internet of Things-based SGs using parameter-tuned bidirectional long short-term memory (PTBiLSTM) with pre-trained feature learning model. The proposed system mainly comprises '4' phases: preprocessing, dataset balancing, feature selection, and ETD. Initially, the consumers' electricity consumption data are collected from the theft detection dataset 2022 (TDD2022) dataset. Then, the data balancing is carried out by using Gaussian distribution, including fuzzy C-means approach to handle the imbalance data. Afterward, the meaningful features from the balanced dataset are extracted using the hard swish and dropout layer included residual neural network-50 (ResNet-50) model. Finally, the ETD is done, which utilizes a PTBiLSTM. The proposed models' performance is evaluated using different performance metrics like accuracy, precision, recall, f-measure, the area under the curve, and kappa. The outcomes proved the efficiency of the proposed method over other related schemes in the ETD of SGs.

Keywords Internet of things · Smart grids · Machine learning · Deep learning · Theft detection dataset · ResNet-50 · Fuzzy C-means · Bidirectional long short-term memory · Electricity theft

1 Introduction

The IoT is a system of interconnected intelligent devices that can exchange data between different sources for various applications. The word "thing" describes an actual object allotted an internet protocol address and the capability to gather and send information through a network without such support of a human or other activity [1]. The IoT is increasingly utilized for intelligent energy monitoring, industrial automation, and other applications. IoT devices are employed in various stages of the smart grid (SG) to track and govern grid statistics to provide reliable and efficient electricity distribution [2]. An SG is an electricity network enhanced with modern digital technology, such as sensors and electronic meters, to improve two-way communication [3]. They allow energy firms to acquire real-time voltage, electricity, active and reactive power, EC, and many other readings from smart meters installed at user residences when coupled with the advanced metering infrastructure (AMI) [4, 5]. Please check the edits made in article title. Technical losses (TLs) and non-technical losses (NTLs) are both engaged in the transmission and distribution of power in an SG [6]. The former includes energy dissipation owing to the Joules action, which is triggered by the electrons being emitted because of heat. The estimate of TLs is required for the tracking of NTLs. ET is a deliberate act of illegal electricity usage, a significant source of NTLs [7, 8]. The ET severely threatens SGs because it

✉ Mahendran Krishnamoorthy
mahae1987@gmail.com

Johny Renoald Albert
jorenoeee@gmail.com

1 Department of ECE, Jansons Institute of Technology, Karumathampatti, Tamil Nadu 641659, India

2 Department of EEE, Erode Sengunthar Engineering College, Perundurai, Tamilnadu 638057, India

causes monetary losses. For example, yearly losses can reach \$6 billion in the US and Canada. Furthermore, because ET overloads the power infrastructure, it hurts its performance [9].

The ET is vulnerable to cyber and physical attacks, in which an illegal client can physically attack the network through bypassing, tampering or hacking with the AMIs, where a third party can attack the system through a cyber-attack [10]. As a result, reducing ET is a central objective for electrical distribution businesses to secure substantial quantities of total energy losses and income [11]. ETD aims to identify unusual behavior in an SG meter's electricity usage (or simply a smart meter). Checking for anomalies in the user's EC patterns can help to identify theft [12]. Furthermore, it is difficult for utilities to discover and confirm ET in residential, commercial, and industrial establishments, rural regions, and big cities via on-site investigations, an ineffective and pricy manual system [13]. In the past, various ML classification algorithms [14], such as support vector machine (SVM), gradient boosting (GB), naive bayes (NB), random forest (RF), decision tree (DT), and others, were used to identify ET. These theft detection techniques have proven to be less expensive [15]. However, only if suitable variables are generated from raw meter readings will the performance of these classification methods be suboptimal [16], and these techniques only consider time-domain features, limiting their performance.

After several decades, much has been accomplished in ML to combat these limitations. DL has been explored in many areas of study due to its capability as a recently introduced part of ML [17]. DL methods are additionally utilized to create models to deal with the enormous quantities of information generated by smart meters [18]. They can learn from vast quantities of information and improve feature extraction and classification processes [19]. Several studies surveyed in the literature includes both ML and DL models [21 to 30] for ET detection in IoT-based smart grids. The problems faced by the existing models are comprehensively studied in Sect. 2.1, that concludes that there is still a lot to be performed to enhance the detection performance of the algorithm. Many vital aspects of a detection algorithm require more subtlety, excluding dealing with missing values, high-dimensional features, and actual malicious behaviors and carrying highly imbalanced datasets. In the existing studies, the convolution neural network (CNN) and long short-term memory (LSTM) combinations of schemes provide better classification outcomes than the other DL and ML models. Because the CNN-based system learns the features automatically without any human interventions, that makes the prediction more accurate. Also, the LSTM is utilized to understand the temporal information from the extracted feature maps and classify the different kinds of thefts more effectively by learning long-term dependencies between the electronic data. Taking this advantage in mind, we are using advanced variants of CNN and LSTM in our work. We are using a modified version of ResNet-50, a pre-trained CNN model that works better than CNN with faster training and BiLSTM, that can read data in both forward and backward directions, producing more accurate results and also being more suitable for large datasets than the LSTM. However, the existing related schemes of ETD using ML and DL frameworks need to be improved in the availability of theft data relative to benign data, a failure to consider dimensionality reduction, the use of standalone (single) ET detectors, etc.

Considering all the above in mind, we develop a novel PTBiLSTM to detect the ET in IoT-based SGs with efficient feature extraction and oversampling techniques. Next, the preprocessing was performed on the dataset. After that, the feature selection is made with the help of a HDResNet-50. The main contributions of the proposed work are as follows:

- To perform efficient preprocessing operations such as missing values imputation, removing outliers and anomalies, and data normalization to enhance the classifier's performance.
- To present an HDResNet-50 model to select essential features from high-dimensional electricity consumer data, thereby improving prediction performance and avoiding over-fitting issues.
- To propose GFCM to solve the data imbalance problem, reducing misclassification errors.
- To propose PTBiLSTM to detect the ET in SGs with chaos and inertia-based moth flame optimization (CIMFO) algorithm-based parameter tuning to avoid higher training time and misleading results.

The remaining phases of the research paper are structured as follows: a survey of recent methodologies related to ET in SG is presented as phase 2. A brief and detailed explanation of the proposed research model is given in phase 3. The implementation outcomes of the proposed method and their comparative analysis with existing works are presented in phase 4. Finally, in phase 5, the conclusion and future studies of the paper are discussed.

2 Related work

This section surveys the recently developed methodologies for ETD in SGs. The techniques are surveyed regarding methods used, results achieved, and the limitations they faced. The solutions drawn by the proposed method to overcome the existing limitations are also given. Asif Nawaz et al. [20] presented an ETD approach in SGs using a convolution neural network (CNN) and extreme GB (XGB). Once collecting data from the dataset, preprocessing was done on the collected dataset using linear interpolation. The essential features from the preprocessed dataset were extracted using CNN, and then the data were classified using the XGB model. The presented scheme achieved an ACY of 92% for ETD when tested on the State Grid Corporation of China (SGCC) dataset. Ashraf Ullah et al. [21] suggested a hybrid DL approach for ETD in SGs. The suggested approach initially collected the data from the SGCC dataset, which was available publicly. The data augmentation was carried out on the collected data to remove redundant and irrelevant data from the dataset. The features were then extracted using CNN. Finally, ET classification was done using the optimized gated recurrent unit (GRU) classifier, in which the parameters of the GRU were optimized using the particle swarm optimization (PSO) algorithm. The method attained an ACY of 93%, which was higher than the existing related schemes.

Farah Mohammad et al. [22] presented an ensemble ML-based classification system for ETD in the SG platform. The suggested model collected the data from the Open Energy Data Initiative (OEDI) dataset. The data cleaning was performed on the collected dataset. The suggested research model used four ML classifiers, such as k-nearest neighbor (KNN), XGB, RF, and multi-layer perceptron (MLP), as an ensemble framework for ETD. The model attained the ACY of ETD between 88–94% for different attacks on known and unknown consumers. Guoying Lin et al. [23] recommended an ETD approach in SGs using under-sampling and re-sampling-based RF (UaRe-RF) and stacked auto-encoder (SAE). The method used two different datasets for data collection, such as the Irish CER Smart Metering Project and actual EC data of unique transformer users in the distribution network of a particular area in China. Once the data were collected, the SAE model was trained to extract the electrical consumption features from the collected data, which were more suitable for theft detection. Finally, UaRe-RF was used to imbalance the dataset and performed the ETD. The method achieved relatively better performance than the existing related schemes.

Salah Zidi et al. [24] suggested ML classifiers for ETD in the SG environment. Once data were collected from the OEDI dataset, the five ML approaches, say DT, KNN, RF, artificial neural network (ANN), and bagging ensemble (BE), were trained to detect the ET in SG. The outcomes showed that the RF classifier attained higher prediction ACY than the other presented schemes. Shoaib Munawar et al. [25] introduced a hybrid classification approach for ETD using bidirectional long short-term memory and bidirectional gated recurrent unit. The method collected the data from the SGCC dataset and then augmented the raw data into a usable format. Then, the class imbalance problem of the augmented dataset was solved using the k-means minority oversampling technique. Then abstract features were extracted using the stochastic feature extraction technique. Finally, the classification was done using a hybrid approach to the extracted feature set. The approach achieved an AUC score of 0.93, which was entirely satisfactory.

Sudeep Tanwar et al. [26] presented a deep ML framework for ETD. Based on smart meter data, the system initially predicted the EC using long short-term memory (LSTM) and a threshold calculator. The predicted EC values were given to the SVM for ETD that classified the energy loss as technical, non-technical, and regular consumption. Md. Nazmul Hasan et al. [27] suggested a CNN-LSTM-based DL for ETD in SGs. The method used the data gathered from the SGCC dataset. The dataset was preprocessed initially, and then the dataset was balanced using a synthetics data generation scheme. Finally, the data were classified using the CNN-LSTM model, which classified the data as normal and ET users with higher ACY.

Sravan Kumar Gunturia and DipuSarkar [28] presented an ETD approach in SGs using an ensemble of ML classifiers. The system consists of three major phases: preprocessing, data imbalance, and classification for ETD. The dataset imbalance problem was solved using the synthetic minority oversampling technique (SMOTE). The balanced data were trained using classifiers such as light boosting, adaptive boosting, extreme boosting, categorical boosting, RF, and extra trees (ETR) to categorize normal and ET users in SG. The results proved that the RF and ETR performed better than the other classifiers with better ACY. Leloko J. Lepolesa et al. [29] suggested a deep neural network (DNN) for ETD in SG. Initially, to extract essential features from the dataset, a minimum redundancy maximum relevance scheme was utilized to extract time and frequency domain features from the dataset. The principal component analysis was used to reduce the dimensions of the extracted feature set. Lastly, the detection of ET was done using DNN. The method attained a higher AUC of 0.90 when tested on the SGCC dataset.

2.1 Problem statement

The survey shows that several machine and DL models of ETD in SGs attained satisfactory performance on the several applied energy consumption datasets. However, they did not provide optimal results for the following reasons:

- The EC data obtained via the dataset are typically extensive, and they are primarily noisy because of daily data recordings. So, it is difficult for the ML models [23, 24, 25, and 29] to handle this massive quantity of data because they use a manual feature extraction process that consumes too much time to process the data. As a result, their performance could be improved to large quantities of time series data.

- Some of the existing works face the dataset imbalance problem in ETD. The dimensionality and diversity of the data are increased because of the tremendous amount of data in the dataset. As a result, a data imbalance problem occurred in the dataset [21, 23, 25, and 27]. Some [22, 24, 26, 28, and 30] use random under-sampling techniques to solve this class imbalance problem. However, they randomly remove the majority of class elements from the dataset and lose the vital information from the dataset, which causes the problem of under-fitting in classification.
- In [28], the authors used SMOTE oversampling approach to get a balanced dataset. However, SMOTE did not consider neighboring examples of the other class labels, increasing the class overlapping problem in the dataset with additional noise. In addition, the approach could be more suitable for practical applications of higher dimensional EC data.
- Some authors presented DL models [21, 22, 26, 27, 28, and 30] to work on the large quantity of data for ETD efficiently. They provide better detection ACY. However, in [21, 25] author used a hybrid DL approach that takes higher execution time compared to a single DL model. Generally, most of the existing works uses CNN for extracting features from the input, but they are computationally expensive and requires large amount of training data.
- In addition, the existing classification schemes fails to focus on parameter tuning, which does not provide the generalization ability of the model that leads to misclassification errors in ETD.
- Most of the works use the SGCC dataset for ETD that only detects normal and theft users, and it does not cover different theft types that occurred in SGs. The methods developed in [22, 24] focused on identifying different electric theft types in SGs. However, they failed to achieve data balancing, dimensionality reduction and parameter tuning in ETD, which degrades the prediction performance of the classifier.

From the survey, it is clear that the existing research is limited to the following issues: data imbalance, parameter tuning, over-fitting, and dimensionality reduction. This motivates us to develop a novel DL approach for ETD with efficient feature extraction and oversampling techniques that provide superior results. The primary goal of the proposed system is given as follows:

- The system proposes GFCM to solve the data imbalance problem in the electronic theft data, which makes the training model easier and prevents the system from being biased toward one class.
- To effectively learn the features from the balanced dataset, a CNN pre-trained model, namely HDResNet-50, is proposed, which learns meaningful information from high-dimensional electricity consumer data and reduces the dimension of the extracted feature maps to a lower dimension using its pooling layers, thereby improving prediction performance and avoiding over-fitting issues. The pre-trained model is better than CNN, which speeds up and simplifies the training process with limited data to converge.
- The system proposes PTBiLSTM to detect the ET in SGs, which helps to learn the temporal information from the extracted feature maps by achieving higher accuracy. The chaos and inertia-based moth flame optimization (CIMFO) algorithm-based parameter tuning is performed to make the model more generalizable by avoiding higher training times and misleading results. Also, our work uses the theft detection dataset 2022 (TDD2022), which contains more ET than previously used datasets. So, this is the first study which identifies different kinds of ETs by addressing all of the problems mentioned above.

3 Proposed methodology

This paper proposes a PTBiLSTM to detect ET in IoT-based SGs. Initially, the data are collected from the publicly available database, namely the TDD2022 for ETD. Then, the collected dataset was preprocessed by performing missing value imputation, removing outliers and anomalies, and data normalization. After that, the dataset imbalance problem is solved by GFCM oversampling technique, which balances the dataset by preventing the network from being toward one class. Then, features from the preprocessed balanced dataset were selected using the HDResNet-50-based convolution neural network that selects the most relevant features for ETD and changes the higher dimension features in the dataset into the lower dimension to reduce over-fitting issues in the classification process. Finally, the classification of the ET is done using the PTBiLSTM network, in which hyper-parameters of the BiLSTM are tuned using the CIMFO. A brief explanation of each of the proposed method's phases is presented in the sub-sections below. Figure 1 shows the structural framework of the proposed research model.

3.1 Preprocessing

Initially, the proposed system collects the data from the TDD2022 dataset containing consumers' EC data. EC data are typically collected via smart meters or other sensors installed at the consumer end. This scenario has a chance of smart meter failure, sensor malfunction, or data transmission

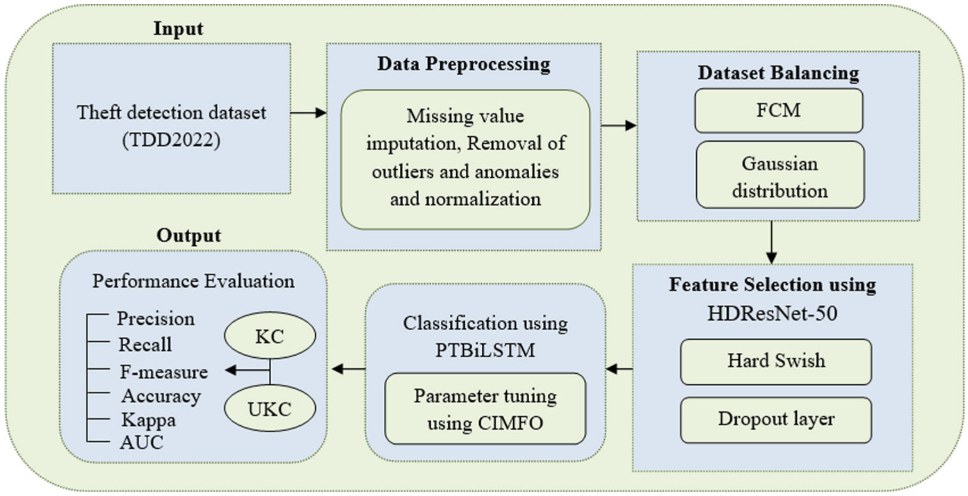


Fig. 1 Structural framework of the proposed research model

and storage server faults. The presence of missing or incorrect data in energy consumption datasets is unavoidable. If those missing cases are discarded, the dataset shrinks significantly, making accurate analysis easy. Furthermore, the collected dataset frequently includes noisy values. Anomalies, missing records, outliers, redundant and overlapping records, inconsistent EC readings, and other types of noise can be found in EC data.

These noises must be managed, or the proposed ETD system provides incorrect predictions and enhances the FPR. The proposed system uses '3' preprocessing techniques to avoid downsizing and handling noises in the dataset. Firstly, outliers or anomalies are dealt with using the three-sigma rule. Secondly, missing values are imputed using the linear interpolation (LI) technique; finally, inconsistent data are managed using the normalization procedure. These stages are briefly described below:

Step 1: Missing values imputation.

The LI method quickly identifies and fills the missing values in the dataset. It uses the average of the following and preceding day's EC values to fill in missing values in the dataset. The process is performed as:

$$\breve{a}''_c = \breve{a}1''_c + \frac{\breve{a}2''_c - \breve{a}1''_c\breve{b}''_c - \breve{b}1''_c}{*}\left(\breve{b}''_c - \breve{b}1''_c\right) \tag{1}$$

where $\breve{a}''_c$ indicates the missing value in EC data at a particular time $\breve{b}''_c$, $\breve{a}2''_c$ refers to the prior value to $\breve{a}''_c$, and $\breve{a}1''_c$ denotes the value that proceeds $\breve{a}''_c$. Moreover,$\breve{b}''_c$, $\breve{b}1''_c$ *and* $\breve{b}2''_c$ shows the time of data for $\breve{a}''_c$, $\breve{a}1''_c$, and $\breve{a}2''_c$, respectively. Suppose if both $\breve{a}1''_c$ and $\breve{a}2''_c$ are non-numeric values in the dataset, then they are replaced by zero; otherwise, missing values are replaced using Eq. (1).

Step 2: Removing outliers and anomalies.

There are multiple anomalies and outliers in the dataset because of data skewing that makes the training process of the system more complex and causes over-fitting issues when performing classification. So, to prevent over-fitting and makes the training process easier, these outliers must be recognized and eliminated. These outliers show the peak electricity consumption that occurs during non-working days. The three-sigma rule identifies and restores anomalies and outliers using the following Eq. (2).

$$\#SigmaRule = \begin{cases} \overset{\cap\cap}{SR}, & if\, \overset{\leftrightarrow}{E}_{(m(n))} > \overset{\cap\cap}{SR} \\ \overset{\leftrightarrow}{E}_{(m(n))}, & otherwise \end{cases} \tag{2}$$

where $\breve{a}\overset{\cap}{R} = AVG\left(\overset{\leftrightarrow}{E}_{(m(n))}\right) + 2\sigma\left(\overset{\leftrightarrow}{E}_{(m(n))}\right)$, σ is the standard deviation, $\overset{\leftrightarrow}{E}_{(m(n))}$ indicates the EC at a current time n , and m refers to the consumer number.

Step 3: Data normalization.

The neural networks are susceptible to diverse data, so the dataset must be standardized or normalized into a specific range once missing values and outliers are removed. A min–max normalization technique was used to scale the dataset according to Eq. (3). The data are scaled from 0 to 1 during the normalization procedure.

$$NT\left(\overset{\leftrightarrow}{E}_{(m(n))}\right) = \frac{\overset{\leftrightarrow}{E}_{(m(n))} - min\left(\overset{\leftrightarrow}{E}\right)}{max\left(\overset{\leftrightarrow}{E}\right) - min\left(\overset{\leftrightarrow}{E}\right)} \tag{3}$$

where $\overset{\leftrightarrow}{E}_{(m(n))}$ indicates the EC at a current time n, m refers to the consumer number, $min\left(\overset{\leftrightarrow}{E}\right)$ denotes the least EC, and $max\left(\overset{\leftrightarrow}{E}\right)$ signifies the highest EC. With this normalization process, the value ranges of the data became more evident.

3.2 Dataset balancing

The typical ETD using ML and DL classifiers mainly faces the class imbalance problem in the collected dataset, because the dataset only contains fewer class samples or labels of ET users than the normal users. So, if the data obtained from the dataset are classified directly through the classifier, it leads to the possibility of identifying ET users as normal. Therefore, the dataset samples must be balanced before proceeding into the classification phase. Generally, many kinds of random oversampling and under-sampling techniques like SMOTE are developed to deal with these issues. However, when applied to the neural networks, they came with the limitations of over-fitting, under-fitting, and computational overhead. So, this paper proposes GFCM to resolve the problem of class imbalance data.

Fuzzy c-means (FCM) clustering is a popular clustering model that groups similar data points in the dataset to form a cluster according to the distance between the data objects by maximizing the objective function. The clustering process is terminated when the maximum number of iterations is reached or when the objective function improvement between two successive iterations is smaller than the minimum amount of improvement specified. It proficiently balances the dataset; however, the size of the minority classes is large, so it takes more time to balance the dataset, and sometimes, an error will happen. Consequently, the proposed system employs the Gaussian distribution strategy, which seeks to achieve balance across all classes by modifying minority class distributions so that the size of their data does not vary markedly compared to the majority class. This Gaussian distribution incorporation in the traditional FCM algorithm is named the GFCM algorithm. The procedures involved in the GFCM algorithm are as follows:

Step 1: Select a minority category from the dataset to be oversampled. The minority class is analyzed further so the dataset can be balanced well before any classification method is implemented.

Step 2: Fed the minority class to the FCM clustering algorithm to partition the minority class into several groups or clusters further based on the underlying clusters and hidden features in the minority class. The goal of FCM here is to maximize the objective function listed below.

$$M^{\cap''}_{h} = \sum_{g=1}^{I} \sum_{v=1}^{V} \chi_{gv}^{h} \| OF_g - y_v \|^2 \tag{4}$$

where $M^{\cap''}_{h}$ denotes the objective function, χ_{gv}^{h} refers to the degree of membership of OF_g in the v^{-th} cluster, OF_g indicates g^{-th} selected feature, y_v signifies the cluster centroid of the v^{-th} cluster, I represents the total selected features from the dataset, and V denotes the number of clusters.

Step 3: Compute the mean (μ) and standard deviation $\left(\sigma^2\right)$ of each feature in the formed clusters using Eqs. (5) and (6).

$$\mu = \frac{\sum_{j=1}^{J} \overset{\cap}{R}_j}{J} \tag{5}$$

$$\sigma = \sqrt{\frac{\sum_{j=1}^{J} \left(\overset{\cap}{R}_j - \overline{\overset{\cap}{R}} \right)^2}{J-1}} \tag{6}$$

where $\overset{\cap}{R}$ denotes the random variable and J indicates the number of elements in the sample.

Step 4: Select the ratio size randomly to generate the number of samples. The best ratio is selected by assigning each instance in the test dataset for the classification. A numerical model is generated to optimize the minority classes' ratio once the sample sizes are selected randomly.

Step 5: Recognize the necessitated samples in each cluster. The samples are selected based on the selected ratio size from step 4 to balance the data in each cluster.

Step 6: Generate synthetic samples in each cluster by applying Gaussian distribution (GD). The GD produces samples based on the mean and variance of the minority class distribution without knowing the actual data distribution. The primary benefit of GD is that it generates synthetic data with a nearly identical probability distribution to the actual minority data.

Step 7: Combine the newly generated synthetic data with the original dataset. Following the random generation of all the necessary samples for balancing data in each cluster, the synthetic data are combined with the raw data to generate a new balanced dataset.

In this way, the GFCM clustering process balances the data samples in the feature-selected dataset. Both theft and normal classes contain an equal number of records fed into the ET identification detection stage.

3.3 Feature selection

After dataset balancing, a feature selection process is carried out to extract more essential features. In most cases, manual feature selection is carried out to extract the higher dimensional user data from the dataset. However, they are tedious and time-consuming and selecting higher dimensional features in smart meters is more challenging. So, this paper proposes a CNN-based pre-trained feature selection scheme, namely HDResNet-50, to effectively select the hidden and dense features from the consumer's profiles. This HDResNet-50 automatically selects the more essential features from the preprocessed dataset and decreases the noise effects in the dataset to a low level.

ResNet-50 has 50 weighted layers, and we extracted features from the last fully connected layer. A residual connection in a layer means that the output of a layer is a convolution of its input plus its input. The traditional ResNet-50 uses the activation function of ReLU for its feature selection processes. However, the neurons of ReLU are fragile, so some of the inputs fall into the challenging saturation area, resulting in irreversible neuronal death and the inability to update the input's weight. Furthermore, the ReLU function sets the portion of the neuron output to zero, which outputs the migration event. Due to this behavior, it is challenging to understand the valuable features. As a result, the classifier's learning performance gets decreased. In addition, extreme sparsity of ReLU leads to higher error rates and a reduction in the model's productive capacity. So, this paper proposes replaces the ReLU with a hard swish (HS) activation function that carefully selects the number of hidden neurons because too many neurons result in over-fitting. In addition, once convolution and pooling operations are performed, a dropout layer (DLY) is included in the fully connected layer of ResNet-50 to prevent the network from over-fitting. These improvements in traditional ResNet-50 using HS and DLY are named HDResNet-50. The architecture diagram of HDResNet-50 is shown in Fig. 2.

The ResNet-50 system works by skipping connections on two to three layers of architectures that incorporate activation functions. Residual blocks on ResNet can be obtained if the input and output data dimensions are equal. The input-balanced data were passed to HDResNet-50 network that consists of 5 stages each of which combines convolution and identity block.

- Stage 1 comprises a 2D convolution with 64 filters with a stride of (2, 2) and a shape size of (7 × 7). The activation function HS and batch normalization complete the channel axis standardization. Finally, a stride of (2, 2) max pooling is included.
- Stage 2 consists of two identity blocks and one 2D convolutional block, each with three filters [64, 64, 256], a kernel size of 3 × 3, and a stride of 1.
- Step 3 is composed of three identity blocks and one convolutional block. Each of these blocks uses three sets of filters [128, 128, 512], has a stride of (2, 2), and has a shape size of (3 × 3).
- Step 4 comprises five identity blocks and one convolutional block. These blocks employ three sets of filters [256, 256, 1024] with a stride of (2, 2) and a size of (3 × 3).
- Stage 5 comprises two identity blocks and one convolutional block. These blocks employ three sets of filters [512, 512, 2048] with a stride of (2, 2) and a shape size of (3 × 3).
- In Stage 6, 2 × 2 average pooling is used. The output is then flattened and sent to the fully connected dropout layer, which reduces the input to the number of classes using a "SoftMax" activation.

The convolution, pooling, and fully connected layers are the components of the ResNet-50 network. The explanation of each component of the network is given below:

3.3.1 Convolution layer

Convolution is the first layer of CNN that extracts relevant features from the input layer by locating local connections among data samples. The convolution operation performed in the network is expressed using Eq. (4).

$$\text{FeatureVector} = \sum \left(\overset{\cap''}{I}_{d \times d} + \overset{\cap''}{W}_{d \times d} \right) + \text{Bias}_{\text{val}} \tag{7}$$

where $\overset{\cap''}{I}_{d \times d}$ refers to the local receptive field of the input data, $\overset{\cap''}{W}_{d \times d}$ indicates the filter weights, $d-$ denotes the kernel size, and Bias_{val} signifies the filter bias of the convolution operation, respectively. The output vector attained using the convolution function is fed into the activation layer for further processes.

3.3.2 Activation function

The activation function is applied to the extracted feature set of the convolution layer. The proposed system uses HS activation function that performs better than ReLU. HS is a novel function closely related to swish. However, it does not fall into the vanishing gradient and saturation problem, and the computation process is also more accessible than other related activation functions. The HS function prevents the weight of the network effectively from changing its value. Thus, the gradient is slightly vanishing, which avoids the problem of network saturation while performing feature selection. It is mathematically expressed as follows:

$$\phi = 2 * f_x * \overset{\cap}{\eta}\left(\alpha_{fx}\right) \tag{8}$$

$$\overset{\cap}{\eta} = \max(0,\ \min(1,\ (f_x\ *\ 0.2 + 0.5))) \tag{9}$$

where f_x indicates a feature vector and α may be a trainable parameter or constant.

3.3.3 Pooling layer

In the max-pooling layer, the feature map obtained via the convolution layer is partitioned into several non-overlapping

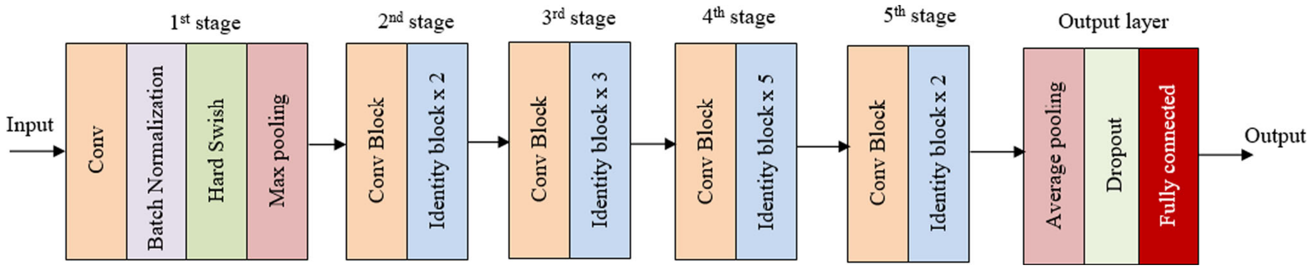


Fig. 2 HDResNet-50 model structure

pooling kernels. From there, the layer considers the maximum value of each pooling kernel and passes it on to the next layer. The primary goal of the pooling layer is to reduce the dimension or size of the convolution data for the next layer. It does not learn anything from the data, but it is used to minimize the computational burden of the network. At the output layer, average pooling is used to determine the average of the elements in the filter-covered feature maps. So, using max pooling, the most prominent features were outputted, and using average pooling, the average of the features was output from the given patch of the feature maps.

3.3.4 Dropout layer

After convolution and pooling operations are completed, a DLY is added to the ResNet-50 network's fully connected layer to protect the network from over-fitting. Dropout is a simple but efficient regularization method for neural networks. The primary benefit of this approach is that it keeps all neurons in a layer from synchronizing their weights. This adaptation, performed in random groups, prevents all neurons from convergent on the same objective, thereby decorrelating the weights. Dropout, in other words, deactivates some neurons in the hidden layers of the network to solve over-fitting and poor generalization problems. Suppose neurons are randomly dropped from the network during training. In that case, other neurons must perform the modeling to forecast the missing neurons, which results in learning numerous independent internal representations in the network. So, the system has become less susceptible to the particular weights of neurons. Consequently, the network is more generalizable and less likely to over-fit the training data.

3.3.5 Fully connected layer

The fully connected layer (FCL) is the final network layer of ResNet-50. It connects the output of the DL to neurons from subsequent layers. Furthermore, it extracts global features from the final feature map obtained from DLY. The mathematical expression of FCL is given in Eq. (10).

$$\overline{\mathrm{FCL}}''_{S_0\times 1} = \overset{\cap}{w}''_{S_0\times S_k} \cdot \overset{\cap}{i}''_{S_i\times 1} + \mathrm{Bias}_{S_0\times 1} \tag{10}$$

where S_o *and* S_k refers to the vector size of the input and output data and $\overline{FCL}''$ indicates the FCL's final output. The obtained final feature $\left(OF_g\right)$ vector using HDResNet-50 is mathematically modeled as follows,

$$OF_g = \{of_1, of_2, of_3, \ldots\ldots..of_I\} \tag{11}$$

where I refers to the total number of optimally selected features from the dataset.

3.4 Electricity theft detection using PTBiLSTM

For the classification of ET users and normal consumers, the proposed system utilizes PTBiLSTM. The bidirectional long short-term memory (BiLSTM) is an improved version of conventional LSTMs that enhances the network's classification performance by extracting hidden time series features from the input data. The BiLSTM combines two LSTM networks in forward and backward directions to perform feature learning effectively. The forward LSTM layer is a forward computation from the start time to the last time, and the backward LSTM layer is a backward computation from the last time to the start time. Each LSTM network contains two hidden layers and one output layer. The outputs of the forward $(\overrightarrow{\overset{\cap}{H}}_t)$ and backward layer $(\overleftarrow{\overset{\cap}{H}}_t)$ at each moment are combined to obtain the final classification output. The BiLSTM effectively classifies normal and theft users in SGs; however, random initialization of weights and bias of BiLSTM causes the network to get stuck into a local point, linear learning, thereby decreasing the prediction ACY. It also increases the execution time of the network when generating random weight and bias values in each iteration. Optimizing the weight and bias values in the network is critical to reducing training errors and improving prediction performance. If the weight and bias of the neurons are optimal, the network output will be more precise.

So, the proposed research model uses the CIMFO algorithm to optimize the weight and bias values in BiLSTM. In addition, the traditional usage of sigmoid activation in the

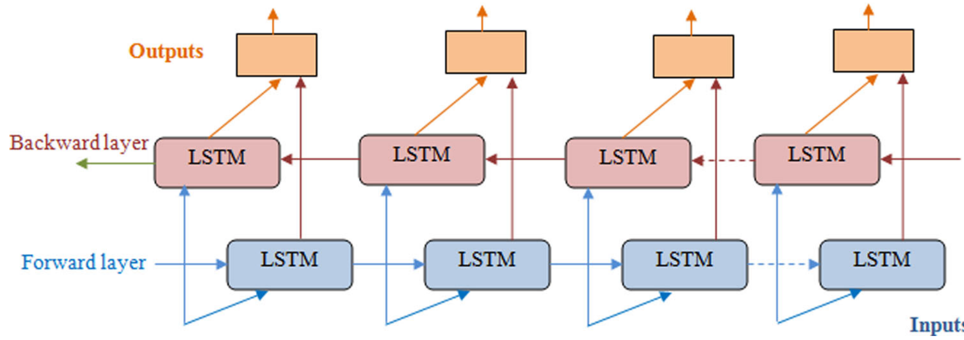


Fig. 3 General diagram of BiLSTM

BiLSTM is replaced with the proposed ER function used in the feature extraction stage, which prevents the network from gradient saturation problems and improves the classification ACY. This parameter tuning in traditional BiLSTM is called PTBiLSTM. The processing steps of BiLSTM are given below, and the architecture of the BiLSTM is shown in Fig. 3.

Step 1: The network has three gates: the input gate $\overset{\cap}{X}''_t$, the forget gate $\overset{\cap}{Y}''_t$, and the output gate $\overset{\cap}{O}''_t$. The forget gate takes the information of the preceding hidden state $\overset{\cap}{H}_{t-1}$ and current input (features of the balanced dataset) $\left(\overset{\cap}{d}_t\right)$ through a point-wise multiplication operation and decides to retain the information from the cell state. By setting the parameters of the three control gates reasonably, the memory function of LSTM can be realized. The core calculation formula is as follows.

$$\overset{\cap}{Y}''_t = \left(\widehat{w}_{\overset{\cap}{Y}''} \cdot \left[\overset{\cap}{H}_{t-1}, \overset{\cap}{d}_t\right] + \widetilde{B}_{\overset{\cap}{Y}''}\right) \tag{12}$$

$$\overset{\cap}{X}''_t = \phi\left(\widehat{w}''_{\overset{\cap}{X}} \cdot \left[\overset{\cap}{H}_{t-1}, \overset{\cap}{d}_t\right] + \widetilde{B}''_{\overset{\cap}{X}}\right) \tag{13}$$

$$\overset{\cap}{O}''_t = \phi\left(\widehat{w}_{\overset{\cap}{O}''} \cdot \left[\overset{\cap}{H}_{t-1}, \overset{\cap}{d}_t\right] + \widetilde{B}_{\overset{\cap}{O}''}\right) \tag{14}$$

where $\widehat{w}_{\overset{\cap}{Y}''}, \widehat{w}_{\overset{\cap}{X}''}, and \widehat{w}_{\overset{\cap}{O}''}$ and $\widetilde{B}_{\overset{\cap}{Y}''}, \widetilde{B}_{\overset{\cap}{dX}}{}'', and \widetilde{B}_{\overset{\cap}{O}''}$ represent the weight and bias values of the forget $\left(\overset{\cap}{Y}''_t\right)$, input $\left(\overset{\cap}{X}''_t\right)$, and output $\left(\overset{\cap}{O}''_t\right)$ gates at the time stamp t, respectively, and ϕ denotes the HS activation function, calculated using Eq. (5). The networks' weight and bias are considered hyper-parameters and randomly chosen between 0 to n-1. In the proposed system, these parameters are tuned using the CIMFO algorithm to obtain optimal weight and bias for BiLSTM to enhance the prediction ACY of the classifier. The weight and bias of the networks are considered as hyper-parameters and they are chosen randomly between the ranges 0 to n-1. In the proposed system, these parameters are tuned using CIMFO algorithm to obtain optimal weight and bias for BiLSTM to enhance the prediction ACY of the classifier.

Step 2: Optimize weight and bias values using CIMFO.

Moth flame optimization (MFO) is a swarm intelligence algorithm that deals with complex real-world optimization issues. It has the advantages of having few setting parameters and being easy to understand and implement. Nonetheless, the conventional MFO suffers from slow convergence and poor ACY. It is also possible to become trapped in a locally optimal solution. Furthermore, MFO and its variations cannot solve higher dimensional optimization issues well. So, a novel improved version of MFO is proposed in this paper with joint search mechanisms to address the shortcomings mentioned above. Firstly, the algorithm employs a logistic tent chaotic map to maintain the diversity of the initial population to attain effective global search. Secondly, a novel adaptive inertia weight function is adopted to improve convergence speed and ACY and escape from the local optimal solutions. These improvements with chaos and inertia mechanism in traditional MFO for enhancing performance are named CIMFO.

Chaotic logistic systems are mathematical systems that define a dynamic deterministic procedure susceptible to initial conditions. They have been used to substitute the random components of the optimization algorithm to improve convergence and mitigate the problem of local minima by moving closer to the optimal solution position. Firstly, the population of moth flame (random weight values and bias) is initialized using a chaotic tent map. It is mathematically expressed as follows:

$$\overset{\cap}{Z}_l^{\tau+1} = \alpha \overset{\cap}{Z}_0 (1 - Z_l), \quad 0 \le Z_0 \le 1 \tag{15}$$

where $\overset{\cap}{Z}_l^{\tau+1}$ indicates the logistic chaos of l^{-th} moth flame, α refers to the constant, and $\overset{\cap}{Z}_0$ denotes the initial moth flame generated randomly between 0 and 1. Next, each individual's fitness $FitNess(Z_l)$ in the population is computed using Eq. (16), which aims to minimize the individuals' mean square error (MSE) to provide a higher classification ACY.

$$\text{FitNess}\left(\overset{\cap}{Z}_l\right) = \min(\text{MSE}) \tag{16}$$

The MSE serves as the simplest form of the loss function. It squares the difference between the predicted output and the ground truth value and averages it across all samples. It is calculated as:

$$\text{MSE} = \sum_{p=1}^{\nu}\left(\overline{\text{RO}}_p^r - \text{OT}_\text{p}^\text{r}\right)^2 \tag{17}$$

where ν refers to the number of samples (number of random weights and bias), $\overline{\text{RO}}_p^r$ indicates the actual output of p^{-th} input unit in cases where the r^{-th} sample is observed in the input, and $\text{OT}_\text{p}^\text{r}$ represents the optimal output of $p^{-\text{th}}$ input unit in cases where the r^{-th} sample is utilized. The individual obtaining lower MSE is chosen as the fittest individuals for the current population. Then, the position updating of the selected individuals is carried out concerning the flame using Eq. (18):

$$\overset{\cap}{Z}_l = \delta\left(\overset{\cap}{Z}_l, \overset{\cap}{T}_e\right) \cdot \kappa_{\text{weight}} \tag{18}$$

where $\overset{\cap}{Z}_l$ denotes the l^{-th} moth, $\overset{\cap}{T}_l$ refers to the e^{-th} flame, and δ signifies the spiral function. The moth's primary update process is a logarithmic spiral function, which is expressed as

$$\delta\left(\overset{\cap}{Z}_l, \overset{\cap}{T}_e\right) = \lambda_l . e^{cs.\beta}\cos(2\pi\beta) + \overset{\cap}{T}_e \tag{19}$$

$$\lambda_l = \left|\overset{\cap}{T}_e - \overset{\cap}{Z}_l\right| \tag{20}$$

where λ_l refers to the distance between the l^{-th} moth and the e^{-th} flame, cs refers to a constant which defines the shape of the spiral, and β signifies a random number. In Eq. (19), κ_{weight} represents adaptive inertia weight that balances the exploration and exploitation capabilities of the MFO rationally, which is estimated as

$$\kappa_{\text{weight}} = \left(\kappa \min_{\max}{}^{e^{(-q/Q)}} + \kappa_{\min}\right) \tag{21}$$

where $\kappa_{\max}$ and $\kappa_{\min}$ represent the maximum and minimum value of inertia weight and $qandQ$ refers to the current and maximum counts of iterations performed by the algorithm. After reducing number of flames in each generation, the corresponding moth updates its position according to the worst flame position.

$$F\text{Num} = \text{Round}\left(\text{Max}F - q * \frac{\text{Max}F - 1}{Q}\right) \tag{22}$$

where Max$F$ indicates the maximum number of flames in the population. Based on the above process, the weight and bias values of the classifier are selected optimally, and the classification process is carried out. The pseudocode of the CIMFO algorithm is shown below

Input: Random weight and bias values of BiLSTM
Output: Optimal value of weight and biases

Begin
  Initialize the MFO population using chaotic logistic method (15)
  For $l = 1\ to\ n$**do**
    Evaluate the fitness of each MFO by using
  End for
  While $q \le Q$**do**
    Update the position of each moth
    Selectlogarithmic spiral function by using

$$\delta(\hat{Z}_l, \hat{T}_e) = \lambda_l . e^{cs.\beta}\, cos(2\pi\beta) + \hat{T}_e$$
$$\lambda_l = |\hat{T}_e - \hat{Z}_l|$$

    Update$FNum$by using the following equation

$$FNum = Round\left(MaxF - q * \frac{MaxF - 1}{Q}\right)$$

  End while
  Return the optimal weight values and biases based on fitness
End

Step 3: Update the cell state information $\left(\overset{\cap}{C}{}''_t\right)$ by performing point-wise multiplication between forget gate's output and the current cell state. The multiplication will result in zero if $\overset{\cap}{Y}''_t$ is 0 which means the previous value's total dropout. Otherwise, it is retained if $\overset{\cap}{Y}''_t$ is 1. It is expressed as follows:

$$\overset{\cap}{C}''_t = \left(\psi_{\overset{\cap}{C}''} \cdot \left[\overset{\cap}{H}_{t-1}, \overset{\cap}{d}_t\right] + \Omega_{\overset{\cap}{C}''}\right) \tag{23}$$

where ψ and Ω refer to the optimal weight and bias values. After that, update the current cell state's information $\left(\overline{\overset{\cap}{C}}''_t\right)$ using point-wise addition operation.

$$\overline{\overset{\cap}{C}}''_t = \left(\overset{\cap}{Y}''_t \times \left(\overset{\cap}{C}''_t\right)\right) + \left(\overset{\cap}{X}''_t \times \left(\overset{\cap}{Y}''_t\right)\right) \tag{24}$$

$$\overset{\cap}{H}_t = \overset{\cap}{O}''_t * \phi\left(\overline{\overset{\cap}{C}}''_t\right) \tag{25}$$

Thus, this hidden layer state $\overset{\cap}{H}_t$ of BiLSTM at the time $t-$ contains forward $\overrightarrow{\overset{\cap}{H}}_t$ and backward $\overleftarrow{\overset{\cap}{H}}_t$ layers. Finally, the proposed PTBiLSTM generates an output vector $\overset{\cap}{G}_t$ by using the following equation,

$$\overset{\cap}{G}_t = \xi\left(\overrightarrow{\overset{\cap}{H}}_t, \overleftarrow{\overset{\cap}{H}}_t\right) \tag{26}$$

where ξ describes the function that combines two output sequences. The function ξ can be a summation, concatenation, multiplication, or average function.

4 Results and discussion

This section presents and discusses the results obtained by the proposed PTBiLSTM for ETD in IoT-based SGs. The proposed method performs training of the ETD system using the TDD2022 dataset. The analysis of the proposed and existing models is also done to prove the performance effectiveness of the proposed approach. The proposed method is implemented in Python with an Intel Core i7-97580H CPU and 16.0 GB of RAM. The detailed explanation is as follows.

4.1 Descriptions of the dataset

The proposed system uses a TDD2022 dataset which is publicly available and is obtained from the OEDI platform. It is a central database of valuable energy research data from the US. The original data include measurements of energy usage of different consumers over a year (12 months), and the readings are collected every hour. The dataset includes

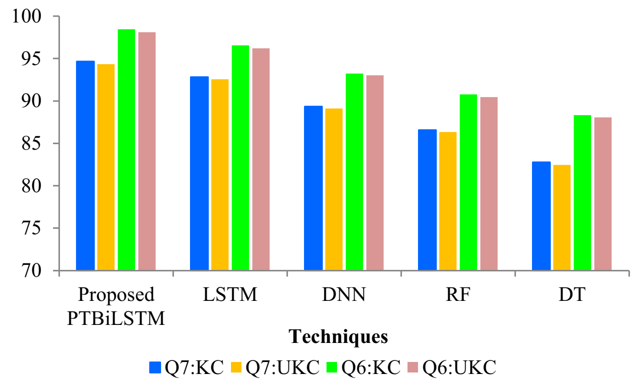


Fig. 4 ACY analysis

sixteen different consumer categories, each with a different energy usage pattern. The collection consists of ten numerical features, one category feature, and 12 characteristics. The dataset has 16 different consumer types, with 35,040 instances of each category. Models with seven output classes and models with six output classes are the two primary mechanisms tested here. Two supporting mechanisms are also evaluated for each primary mechanism. The first, referred to as a "known consumer," used input, consumption, and consumer-type attributes. The second option, "unknown consumer," only employs consumption attributes; the list of features omits the consumer type in this case. They are "7 classes (Q7)—known consumer (KC)," "7 classes (Q7)—unknown consumer (UKC)," "6 classes (Q6)—known consumer (KC)," and "6 classes (Q6)—unknown consumer (UKC)."

4.2 Performance analysis of the proposed PTBiLSTM

The performance effectiveness of the proposed PTBiLSTM is weighted against the existing deep neural network (DNN), LSTM, RF, and DT algorithms. The proposed system uses some commonly used metrics, namely, ACY, PN, RL, FE, AUC, and kappa metrics, to analyze the model's performance.

4.3 Accuracy analysis

The percentage of accurate forecasts over all other predictions is known as ACY. Here, the results of proposed and existing classifiers are compared for the several classes of known and unknown consumers, which is shown in Fig. 4. Figure 4 indicates that the proposed PTBiLSTM achieves a higher level of ACY than the conventional methodologies. Deeply, for Q7, the existing DT offers 82.78% of ACY in KC and 82.46% of ACY in UKC categories, and for Q6, the existing DT achieves 88.32% of ACY for known consumers and 88.07% of ACY for unknown consumers. Comparing all methods, the DT model obtains the lower performance for ETD. However, the proposed one achieves ACY of 94.67% for KC with Q7, 94.34% for UKC with Q7, 98.47% for KC with Q6, and 98.12% for UKC with Q6, which is higher when compared to the existing DT. Also, the proposed system attained higher ACY than the existing LSTM, DNN, and RF classifiers. Thus, results based on ACY show that the proposed work is very competitive against other methods recently undertaken.

4.3.1 Precision and recall analysis

Here, the PN and RL of the classifiers for ETD are carried out, and the results are tabulated in Table 1. The results indicate that the proposed PTBiLSTM model outperforms the existing LSTM, DNN, RF, and DT classifiers by obtaining higher values of PN and RL for ETD. In Table 1(a), the PN results of the classifiers are tabulated. The PN of an algorithm may be determined by dividing the number of accurate predictions by the total amount of positive outcomes. For Q7 classes, the PN attained by the conventional LSTM, DNN, RF, and DT is 92.42%, 88.94%, 86.16%, and 82.38% for KC and 92.17, 88.81, 85.94, and 82.08% for UKC, respectively, while the proposed PTBiLSTM attained better PN of 94.27% for KC and 93.94% for UKC, respectively.

In addition, for six classes (Q6), the PN achieved by the conventional LSTM, DNN, RF, and DT are 96.16, 92.83, 90.38, and 87.92% for KC and 97.01, 92.73, 90.05, and 87.77% for UKC, respectively, while the proposed PTBiLSTM attained better PN of 98.07% for KC and 97.72% for UKC, respectively. These results show that the proposed PTBiLSTM performs better than the existing methods. Next, Table 1(b) compares the RL results obtained by the classifiers. The ratio of true positive measures to all actual measures is called RL. When considering the RL metric, the proposed PTBiLSTM offers higher RL than the conventional methods. While analyzing the results of seven output classes (Q7), the proposed PTBiLSTM attained the highest results with an RL rate of 94.87% for KC and 94.54% for UKC; likewise, analyzing the results with the six output classes (Q6), the proposed one also achieves higher RL, 98.67% for KC and 98.32% for UKC, respectively. This is higher RL than the traditional methods, showing the efficacy of the proposed one for ETD.

4.3.2 Analysis based on F-measure

Figure 5 shows the FE analysis of the proposed and existing classifiers, a harmonic mean of the ratio of PN and RL. The proposed one obtained better FE values for both Q7 and Q6 classes than the conventional methods. The maximum FE values obtained by the existing LSTM for both KC and UKC in the Q7 classes are 92.72 and 92.47%, and in the Q6 classes, they are 96.46 and 96.06%. However, the proposed method beats this model by improving the FE value up to 94.65 and 94.24% for KC and UKC in Q7 class and 98.29 and 98.02% for KC and UKC in Q6 class. Likewise, when comparing the FE results of the other classifiers with the proposed method, the proposed PTBiSLTM obtains remarkable performance for ETD for both classes of known and unknown consumers.

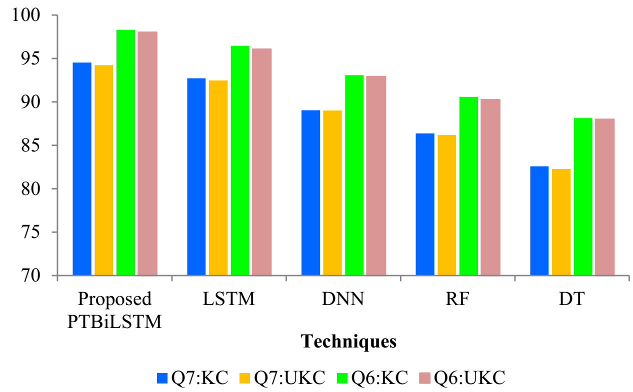


Fig. 5 F-measure analysis

Analysis based on AUC and Kappa In this section, the outcomes of the proposed and existing classifiers are analyzed in terms of AUC and Kappa metrics, which are given in Table 2. Table 2(a) shows the AUC comparison and Table 2(b) shows the kappa comparison of the models. A high AUC for a model suggests a better capacity to predict classes. The AUC can be between 0 and 1, with a value of 0 indicating that the classifier misclassifies positive data as negative and a value of 1 showing that it correctly distinguishes between positive and negative data. From the table, it is known that the outcomes of the proposed method give better performance than the conventional methods. Here, the proposed PTBiLSTM has AUC of 95.67% with Q7 classes including KC, 95.34% with Q7 classes including UKC, 99.47% with Q6 classes including KC, and 99.12% with Q6 classes including UKC, respectively, which is higher than the convention methods.

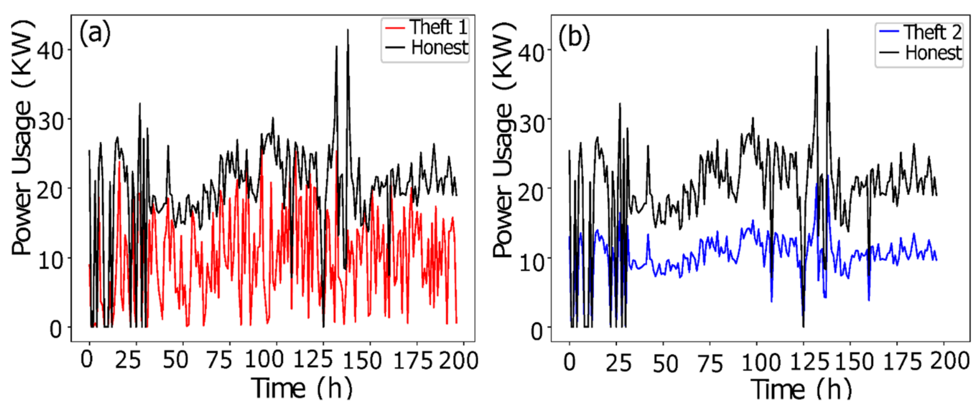
Since the kappa considers the possibility of coincidental agreement, it is generally believed to be more trustworthy than basic ACY. If the classification is perfect, kappa's percentage value is 100; if it is merely the result of coincidence, kappa is equal to zero. The existing RF classifier has kappa of 84.78% with Q7 classes KC consumer and 93.45% with six classes including UKC, respectively, which is higher than the conventional methods. However, the proposed PTBiLSTM has kappa of 92.87% with Q7 classes including KC, 92.56% with Q7 classes including UKC, 91.47% with Q6 classes including KC, and 91.12% with Q6 classes including UKC, respectively, which is higher than the convention methods. Also, the proposed one has a higher kappa score than the

Table 1 Precision and recall analysis of the classifiers

Techniques	Q7: KC	Q7: UKC	Q6: KC	Q6: UKC
(a)				
Proposed PTBiLSTM	94.27	93.94	98.07	97.72
LSTM	92.42	92.17	96.16	97.01
DNN	88.94	88.81	92.83	92.73
RF	86.16	85.94	90.38	90.05
DT	82.38	82.08	87.92	87.77
(b)				
Proposed PTBiLSTM	94.87	94.54	98.67	98.32
LSTM	93.02	92.77	96.76	96.41
DNN	89.54	89.31	93.53	93.33
RF	86.76	86.54	90.98	90.65
DT	82.98	82.66	88.52	88.37

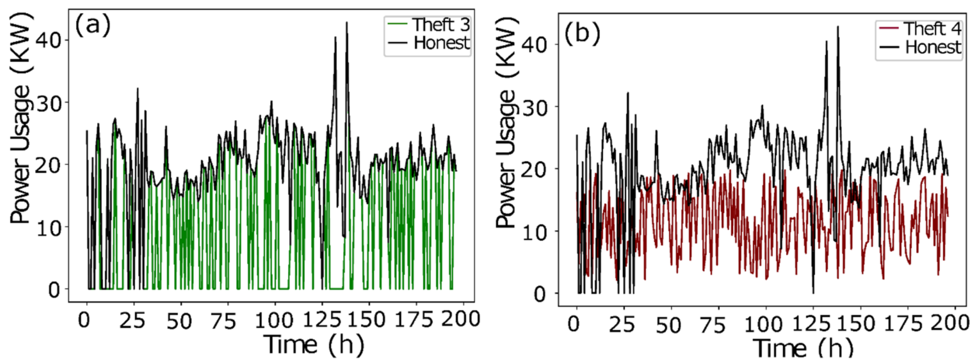
Table 2 AUC and Kappa analysis

Techniques	Q7: KC	Q7: UKC	Q6: KC	Q6: UKC
(a)				
Proposed PTBiLSTM	95.67	95.34	99.47	99.12
LSTM	93.82	93.57	97.56	97.21
DNN	90.34	90.11	94.23	94.03
RF	87.56	87.34	91.78	91.45
DT	83.78	83.46	89.32	89.07
(b)				
Proposed PTBiLSTM	92.87	92.56	91.47	91.12
LSTM	90.64	90.45	89.56	89.21
DNN	87.67	87.34	86.23	86.03
RF	84.78	84.78	83.78	83.45
DT	82.34	82.34	81.32	81.07



(a) **(b)**

Fig. 6 Proposed case 1 and 2



(a) (b)

Fig. 7 Proposed case 3 and 4

existing LSTM, DNN, and DT. Hence, it is evident that the proposed model is superior to other existing methodologies.

Figures 6 and 7 show the theft analyzation of the proposes system for the energy usage of the customers. The thefts and honest consumers are identified based on their power usage over a period of time. The theft users modify the data from honest users by multiplying a random integer with the actual data which results in discontinuity in user's data consumption. Figure 7 displays that data manipulation of the users when 1 or 0 is multiplied at random with time series data, the result is either the initial consumption or a consumption of exactly 0. Between 0 and 1, there is no ramping function with that fully connected grid load or the cut-off simple ON/OFF switching action. Totally, four theft users are identified for the given period of time with the power usage levels from 0 to 40KW. The outcomes clearly indicate that the proposed system more accurately detects the different kinds of ETs with minimal loss compared to existing algorithms.

5 Conclusion

This paper proposed a novel PTBiLSTM-based DL approach to detect ET in IoT-based SGs, which follow four phases such as preprocessing, data balancing, feature selection, and classification to perform its processes. The proposed system uses the TDD2022 dataset to analyze the effectiveness of the system. The outcomes of the proposed PTBiLSTM are compared to the existing LSTM, DNN, RF, and DT classifiers concerning the ACY, PN, RL, FE, AUC, and kappa metrics. The classifiers are evaluated based on the following classification modules: Q7 classes with known and unknown consumers and Q6 classes with known and unknown consumers. The results show that the proposed PTBiLSTM performs better than the existing schemes in all known and unknown consumers. The PTBiLSTM obtained the highest ACY of 94.67, 94.34, 98.47, and 98.12% with Q7 classes of KC, Q7 classes of UKC, Q6 classes of KC, and Q6 classes of UKC, respectively. These best results of PTBiLSTM for ETD are because of efficient schemes of preprocessing, feature selection, oversampling, and classification. The efficient methods used in all phases of the proposed model overcome the problem of higher dimensionality, over-fitting, data imbalance, and lower prediction ACY in existing schemes of ETD. So, from the result analysis, it is concluded that the proposed PTBiLSTM efficiently detects the various types of ET in IoT-based SGs with higher prediction ACY and lower computational overhead than existing related schemes. In the future, the work will be extended to identify different kinds of intrusions that occurred in IoT-based SGs networks. Some cryptographic approaches will be used to prevent the network from intrusions.

Author's contribution Authors 1 wrote the paper. Authors 2 collected the data and performed the analysis. All authors reviewed the manuscript.

Funding The author did not receive support from any organization for the submitted work. No funding for this study. The authors have no relevant financial or non-financial interests to disclose.

Declarations

Conflict of interest The author has no relevant financial or non-financial interests to disclose.

Ethics approval The paper is an original contribution of research and is not published elsewhere in any form or language.

Humans and animal rights No Humans or Animals were involved in the experimentation.

References

1. Barman BK, Yadav SN, Kumar S, Gope S (2018) IOT based smart energy meter for efficient energy utilization in smart grid. In: 2018 2nd Interational conference on power, energy and environment: towards smart technology (ICEPE) (pp 1–5). IEEE
2. Khan F, Siddiqui MAB, Rehman AU, Khan J, Asad MTSA, Asad A (2020). IoT based power monitoring system for smart grid

applications. In 2020 International conference on engineering and emerging technologies (ICEET) (pp 1–5). IEEE
3. Khan ZA, Adil M, Javaid N, Saqib MN, Shafiq M, Choi JG (2020) Electricity theft detection using supervised learning techniques on smart meter data. Sustainability 12(19):8023
4. Kim JY, Hwang YM, Sun YG, Sim I, Kim DI, Wang X (2019) Detection for non-technical loss by smart energy theft with intermediate monitor meter in smart grid. IEEE Access 7:129043–129053
5. Yao D, Wen M, Liang X, Fu Z, Zhang K, Yang B (2019) Energy theft detection with energy privacy preservation in the smart grid. IEEE Internet Things J 6(5):7659–7669
6. 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
7. Javaid N, Jan N, Javed MU (2021) An adaptive synthesis to handle imbalanced big data with deep siamese network for electricity theft detection in smart grids. J Parall Distributed Comput 153:44–52
8. Shehzad F, Javaid N, Almogren A, Ahmed A, Gulfam SM, Radwan A (2021) A robust hybrid deep learning model for detection of non-technical losses to secure smart grids. IEEE Access 9:128663–128678
9. Takiddin A, Ismail M, Zafar U, Serpedin E (2020) Robust electricity theft detection against data poisoning attacks in smart grids. IEEE Trans Smart Grid 12(3):2675–2684
10. Rouzbahani HM, Karimipour H, Lei L (2020) An ensemble deep convolutional neural network model for electricity theft detection in smart grids. In:2020 IEEE international conference on systems, man, and cybernetics (SMC) (pp 3637–3642). IEEE
11. Aslam Z, Javaid N, Ahmad A, Ahmed A, Gulfam SM (2020) A combined deep learning and ensemble learning methodology to avoid electricity theft in smart grids. Energies 13(21):5599
12. Punmiya R, Choe S (2019) Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing. IEEE Trans Smart Grid 10(2):2326–2329
13. Kulkarni Y, Hussain S, Ramamritham K, Somu N (2021) EnsembleNTLDetect: an intelligent framework for electricity theft detection in smart grid. In: 2021 International conference on data mining workshops (ICDMW) (pp 527–536). IEEE
14. Barolli L, Terzo O (2020) Complex, intelligent, and software intensive systems. Springer International Publishing
15. Bohani FA, Suliman A, Saripuddin M, Sameon SS, Md Salleh NS, Nazeri S (2021) A comprehensive analysis of supervised learning techniques for electricity theft detection. J Electrical Comput Eng, 2021
16. Razavi R, Gharipour A, Fleury M, Akpan IJ (2019) A practical feature-engineering framework for electricity theft detection in smart grids. Appl Energy 238:481–494
17. Review of load data analytics using deep learning in smart grids: open load datasets, methodologies, and application challenges
18. Kong X, Zhao X, Liu C, Li Q, Dong D, Li Y (2021) Electricity theft detection in low-voltage stations based on similarity measure and DT-KSVM. Int J Electr Power Energy Syst 125:106544
19. Adil M, Javaid N, Qasim U, Ullah I, Shafiq M, Choi JG (2020) LSTM and bat-based RUSBoost approach for electricity theft detection. Appl Sci 10(12):4378
20. Nawaz A, Ali T, Mustafa G, Rehman SU, Rashid MR (2023) A novel technique for detecting electricity theft in secure smart grids using CNN and XG-boost. Intell Syst Appl 17:200168
21. Ullah A, Javaid N, Yahaya AS, Sultana T, Al-Zahrani FA, Zaman F (2021) A hybrid deep neural network for electricity theft detection using intelligent antenna-based smart meters. Wirel Commun Mob Comput 2021:1–19
22. Mohammad F, Saleem K, Al-Muhtadi J (2023) Ensemble-Learning-based decision support system for energy-theft detection in smart-grid environment. Energies 16(4):1907
23. Lin G, Feng X, Guo W, Cui X, Liu S, Jin W, Ding Y (2021) Electricity theft detection based on stacked autoencoder and the undersampling and resampling based random forest algorithm. IEEE Access 9:124044–124058
24. Zidi S, Mihoub A, Qaisar SM, Krichen M, Al-Haija QA (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
25. Munawar S, Javaid N, Khan ZA, Chaudhary NI, Raja MAZ, Milyani AH, Ahmed AA (2022) Electricity theft detection in smart grids using a hybrid BiGRU–BiLSTM model with feature engineering-based preprocessing. Sensors 22(20):7818
26. Tanwar S, Kumari A, Vekaria D, Raboaca MS, Alqahtani F, Tolba A, Sharma R (2022) GrAb: a deep learning-based data-driven analytics scheme for energy theft detection. Sensors, 22(11):4048
27. Hasan MN, Toma RN, Nahid AA, Islam MM, Kim JM (2019) Electricity theft detection in smart grid systems: A CNN-LSTM based approach. Energies 12(17):3310
28. Gunturi SK, Sarkar D (2021) Ensemble machine learning models for the detection of energy theft. Electric Power Syst Res 192:106904
29. Lepolesa LJ, Achari S, Cheng L (2022) Electricity theft detection in smart grids based on deep neural network. IEEE Access 10:39638–39655