sustainability

MDPI

Article

A Hybrid Deep Learning-Based Model for Detection of Electricity Losses Using Big Data in Power Systems

Adnan Khattak [1], Rasool Bukhsh [2,*], Sheraz Aslam [3,*], Ayman Yafoz [4], Omar Alghushairy [5] and Raed Alsini [4]

[1] Department of Computer Science, Abasyn University, Islamabad 44000, Pakistan
[2] Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[3] Department of Electrical Engineering, Computer Engineering, and Informatics, Cyprus University of Technology, Limassol 3036, Cyprus
[4] Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[5] Department of Information Systems and Technology, College of Computer Science and Engineering, University of Jeddah, Jeddah 21589, Saudi Arabia
* Correspondence: rasoolbax.rb@gmail.com (R.B.); sheraz.aslam@cut.ac.cy (S.A.)

Abstract: Electricity theft harms smart grids and results in huge revenue losses for electric companies. Deep learning (DL), machine learning (ML), and statistical methods have been used in recent research studies to detect anomalies and illegal patterns in electricity consumption (EC) data collected by smart meters. In this paper, we propose a hybrid DL model for detecting theft activity in EC data. The model combines both a gated recurrent unit (GRU) and a convolutional neural network (CNN). The model distinguishes between legitimate and malicious EC patterns. GRU layers are used to extract temporal patterns, while the CNN is used to retrieve optimal abstract or latent patterns from EC data. Moreover, imbalance of data classes negatively affects the consistency of ML and DL. In this paper, an adaptive synthetic (ADASYN) method and TomekLinks are used to deal with the imbalance of data classes. In addition, the performance of the hybrid model is evaluated using a real-time EC dataset from the State Grid Corporation of China (SGCC). The proposed algorithm is computationally expensive, but on the other hand, it provides higher accuracy than the other algorithms used for comparison. With more and more computational resources available nowadays, researchers are focusing on algorithms that provide better efficiency in the face of widespread data. Various performance metrics such as F1-score, precision, recall, accuracy, and false positive rate are used to investigate the effectiveness of the hybrid DL model. The proposed model outperforms its counterparts with 0.985 Precision–Recall Area Under Curve (PR-AUC) and 0.987 Receiver Operating Characteristic Area Under Curve (ROC-AUC) for the data of EC.

Keywords: class imbalance; gated recurrent units; convolutional neural network; electricity theft detection; non-technical losses; smart grids

Citation: Khattak, A.; Bukhsh, R.; Aslam, S.; Yafoz, A.; Alghushairy, O.; Alsini, R. A Hybrid Deep Learning-Based Model for Detection of Electricity Losses Using Big Data in Power Systems. *Sustainability* **2022**, *14*, 13627. https://doi.org/10.3390/su142013627

Academic Editor: Andreas Kanavos

Received: 8 August 2022
Accepted: 5 October 2022
Published: 21 October 2022

1. Introduction

Electricity has become a basic need in the modern world, as it is used in homes, businesses, and industry. To distribute electricity to these sectors, a network is formed, which is called the power grid. Technically, the power grid consists of a production side and a demand side. Electricity generation is increased or decreased depending on the demand side's needs. Unfortunately, some of the electricity produced is lost during generation, transmission, and distribution. Energy losses are divided into two main classes: non-technical losses (NTL) and technical losses. Various methods, techniques, and tools are in practice or are proposed to address technical losses.

On the demand side, one of the NTLs is electricity theft. Electricity loss is a major issue for power utility companies, as it causes major disruption to their operations, which leads

to loss of revenue, increased generation load, and excessive electricity bills for legitimate consumers. Moreover, electricity loss also causes issues related to economic growth and power infrastructure stability. NTL, also known as commercial losses, happen mostly due to electricity theft and fraud. Power utility companies still lose large amounts of revenue due to unlawful electricity theft and fraud by electricity consumers. This theft places a heavy burden on the power grid infrastructure and results in fires that threaten public safety. They also cause loss of revenue for electrical generation companies [1–3]. It is a challenge to address power caused by theft. Theft can be done by tampering with electricity meters, double-tapping attacks, changing meter readings through communication links, and using shunt devices. It is an open secret that power utilization is strongly connected with the development of a country and is hence a vital measure that shapes the foundation of industrialization. With the consistently increasing need for power usage, electricity theft is at a peak. Fossil fuel combustion from electricity generation causes 70% of greenhouse gas (GHG) emissions [4]. In spite of endeavors to reduce GHG outflows, electricity theft overshadows these endeavors in developing countries. The capacity to create electric power is diminished as a result of resources lost to energy theft. Due to electricity theft, unnecessary blackouts/load-shedding occur, which encourages users to opt alternative energy resources to fulfill their requirements, including using petrol and diesel generators that cause GHG emissions.

The majority of climate talks have focused on how to lower GHG emissions; very few have examined the consequences of energy theft. By continuously monitoring the electrical system and isolating energy-theft hotspots from a distance, Smart Meters (SM) are suggested as a strategy to prevent energy theft. All transformers, distribution poles, and customer houses should have SMs. The measurements are subsequently transmitted over a communication network to the distribution company's database for examination, and if trouble areas are found, power is cut off remotely. This technology would enhance performance, which would immediately result in a decrease in GHG emissions while also increasing total returns to the distribution firm. It would also promote transparency in the metering process.

Moreover, NTLs cause USD 75 billion in lost revenue in the United States. This amount is enough to power 77,000 households for a year [5]. A World Bank report shows that China, Brazil, and India suffer 16%, 25%, and 6% losses in electricity supply, respectively [6]. According to Joker et al. [7] such losses are not only limited to developing countries; developed countries such as the U.S. and the U.K. bear losses of USD 6 billion and GBP 173 million, respectively, each year. The above discussion shows that an efficient electricity theft detection (ETD) model is required to detect NTLs. In the literature, hardware devices, and data-driven and game-theoretic approaches are used to detect NTLs. Hardware-based approaches use sensors and radio identification tags to distinguish between honest and malicious samples. However, these approaches are expensive, require huge maintenance costs, and do not provide optimal results under extreme weather conditions [3,8–10]. Methods based on game theory design a utility function among electric utilities, stakeholders, and customers. However, it is difficult to implement an accurate utility function. Moreover, these approaches are less accurate and have a high false-positive rate (FPR) [11–14].

The introduction of smart power grids opens new opportunities for ETD. A smart grid is an upgraded version of a conventional power grid and consists of smart meters, sensors, and computing devices that have self-healing mechanisms and communication technologies. The smart meters and sensors obtain data on consumers' electricity consumption (EC), electricity prices, and the status of the grid infrastructure [15,16]. The data-driven approaches are trained on the collected EC data to distinguish between honest and malicious samples. These approaches have received a lot of focus from the research community, but they have the following limitations: curse of dimensionality, class imbalance problems, and low detection rates for standalone ML and DL models. Moreover, conventional ML models such as k-nearest neighbors and naïve Bayes have high FPRs. As mentioned in the

literature, electric utilities cannot tolerate low detection rates and high FPRs because for on-site inspection they have limited resources.

This paper presents a hybrid DL model (named HGC) that is a combination of a gated recurrent unit (GRU) and a convolutional neural network (CNN). GRU extracts temporal features, while CNN retrieves abstract patterns from EC data. The advantages of the models are summarized in the HGC model. It also outperforms existing models. The uneven distribution of class patterns leads to poor performance. This problem leads to majority class bias, which leads to incorrect results. In this paper, a hybrid approach consisting of undersampling and oversampling methods is presented to deal with the uneven distribution of class samples. The main contributions of the paper are listed below.

- We present an HGC model that combines the advantages of GRU and CNN. It is the first study that combines the advantages of sequential and non-sequential models.
- A CNN model extracts latent or abstract patterns, while a GRU retrieves temporal patterns from EC data. The curse of dimensionality is addressed with both DL models.
- The adaptive approach of synthetic minority oversampling and TomekLinks are used to discuss the problem of class imbalance.
- The performance of the HGC model is evaluated using a real EC dataset obtained from the State Grid Corporation of China (SGCC).
- To verify the real efficiency of the proposed model, extensive experimentation is performed based on recall, accuracy, precision, F1 score and FPR.

The rest of the paper is organized as follows. Section 2 presents an overview of related literature. We present the Problem Statement in Section 3, followed by Materials and Methods in Section 4. The Proposed Model is outlined in Section 5. Section 6 contains the Experimental Analysis and Discussion. The Experimental Outcome and Arguments are discussed in Section 7. Finally, we come to an end in Section 8.

2. Related Literature

The tools and techniques proposed in the literature to detect NTLs are studied in this part of the document. In [5], a model combining CNN and multilayer perceptron (MLP) is used. It integrates the advantages of both DL models, which is why it gives better results than standalone models. The first model is employed to extract hidden, abstract patterns, while the latter one is used for extracting meaningful information. The class imbalance problem, however, is not addressed, which makes the ML and DL models biased towards majority class samples and ignore minority ones. Moreover, MLP does not give results on sequential datasets. Joker et al. [7] propose an electricity theft detector that is developed using an SVM classifier to differentiate between malicious and honest customers. It is the first study that integrates a ML model and hardware devices to capture drift changes in data that can happen due to many reasons: e.g., a different number of members in a household or weather changes. Some authors utilize random undersampling to solve the uneven distribution of class samples. However, this technique creates underfitting. Moreover, they utilize hardware devices that make the proposed solution expensive. In [17], the authors propose a theft detector that contains gradient boosting classifiers. The authors introduce the concept of stochastic features, which enhance the detection rate and reduce the FPR. Moreover, they conduct a comparative study and prove that boosting classifiers perform better than SVM on an Irish dataset. Moreover, electricity theft cases are updated by arguing that existing theft cases' resemblance to real-time samples is the least. Random oversampling is employed to handle the uneven distribution of class samples, which creates an overfitting problem. The curse of dimensionality is a big nuisance and reduces the detection-rate of ML and DL models. In [18], the authors use heuristic techniques to select optimal combination of features from EC data, which solves overfitting, memory constraints, and computational overhead issues. However, they use accuracy as a fitness function to evaluate the efficacy of meta-heuristic techniques, which is not a good practice.

In [19], a long short-term memory (LSTM)-dependent framework is suggested. It is proposed for differentiating between malicious and normal patterns as well as changes

due to drift. Based on our knowledge, this is the first study that considers drift changes with malicious patterns and reduces FPR. The Power utilities are unable to bear high FPR due to their limited resources to inspect on the site. Fenza et al. [20] propose a model that integrates the benefits of both CNN and random forest. The former is used to obtain abstract features, while the latter is used to differentiate malicious and normal patterns in EC data. The class imbalance problem is handled using SMOTE, which creates overfitting. In [21], a DL model is proposed that integrates the benefits of both LSTM and MLP. This is the first article that has leveraged the benefits of both sequential and non-sequential data. The class imbalance problem is not considered, which is why ML and DL models give biased results. In [22], an ensemble deep CNN is used for detection of atypical behaviors in EC data. Imbalanced data are a severe issue in ETD and is handled through random bagging. Finally, a well-known voting ensemble strategy is utilized to decide between malicious and normal patterns. Ghori et al. [23] conduct a comparison study between different conventional ML classifiers using a real EC dataset. The ANN and boosting classifiers such as LightBoost, CatBoost, and XGBoost give better performance than other models. Moreover, the curse of dimensionality is dealt with by selecting optimal combination features.

In [24], the authors put forward a fascinating technique for NTL detection using smart meter data. Moreover, auxiliary information is utilized to enhance the accuracy of ML models. Different features are built using distance and density outlier-detection methods. The proposed model is employed in smart grids to distinguish illegitimate patterns from legitimate patterns. In [25], Hasan et al. put forward the idea of identifying low-voltage stations and comparing the performance of supervised and unsupervised learning methods. The suggested method gives better results in contrast to SVM and DT-SVM.

Ismail et al. [26], merge the integrated model of CNN and LSTM. This is the first study that integrates the benefits of both DL learning models. Moreover, the uneven distribution of class samples is another severe issue. SMOTE is utilized to handle this issue. The proposed hybrid model achieves 89% accuracy, which is more than conventional ML and DL models.

The poisoning attack problem in smart grids is proposed by Maamar et al. [27]. They introduce a sequential and parallel DL-based autoencoder based on GRU and LSTM models. The deep neural network performs better than a shallow neural network. In [28], it is revealed that existing studies mostly monitor attacks on the consumer side. No one focuses on the distribution side, where hackers hack utility meters and create higher electricity bills. In their study, they introduce a hybrid C-RNN-based model and prove that it performs well compared to other DL models. The proposed model is evaluated on SCADA meter readings.

In [29], a new hybrid approach is introduced that integrates the benefits of k-mean clustering and a deep neural network. Irish Smart Energy Trials data are used for model evaluation. However, if the authors utilize other advanced clustering algorithms, then proposed model increases the performance. Shehzad et al. [30] introduce a smart system for ETD. The system integrates the benefits of statistical methods and different DL models such as MLP, LSTM, RNN, and GRU. The proposed technique is evaluated on real data from Singaporean homes. However, the performance of the suggested technique is not checked using other performance measures such as F1-score, recall, precision, FPR, ROC-AUC, and PR-AUC.

3. Problem Statement

In [3], the authors propose a theft detector consisting of an SVM to discriminate between malicious and normal samples. However, they do not use a feature selection or extraction approach to deal with the curse of dimensionality. Overfitting leads to high accuracy when using training data compared to test data when ML and DL models are used. Moreover, in [17], the black-hole algorithm (BHA) is used to handle the high dimensional data. BHA is a meta-heuristic method that requires high and complex computations to find an optimal feature combination with which ML models achieve better results. For

this reason, it is not suitable for real-time smart-grid applications. Moreover, the problem of class imbalance is another serious problem in ETD. There are more samples of normal classes than malicious classes. Zheng et al. [1] do not use any approach to solve this problem. In [8], SMOTE is used to improve the minority class samples. However, with this approach, there is a tendency for the ML or DL models to run into an overfitting problem as sample size increases. In [3], a random undersampling approach is employed to compensate for the unequal distribution of normal and malicious samples. However, this approach removes important information and creates the problem of underfitting. In the literature, authors usually use conventional ML models such as SVM, DT, and NB. These models have low detection rates and high FPRs. Therefore, an efficient framework with accurate identification of NTLs in EC needs to be proposed.

4. Materials and Methods

Section 4.1 is about acquiring the dataset; data preprocessing is covered in Section 4.2, which include handling missing values, removing outliers, normalizing data values, and class imbalance problems; and in Section 5 the proposed model is discussed.

4.1. Acquiring the Dataset

In this study, to appraise the performance of the suggested model, data from the State Grid Corporation of China (SGCC) are used, as it is the only publicly available dataset; it includes 42,372 records of consumers, 3615 of which are thieves, while the rest are ordinary consumers (https://github.com/henryRDlab/ElectricityTheftDetection (accessed on 2 March 2022)). Each consumer has a label, either 1 or 0, where 0 represents a normal consumer, and 1 represents a malicious consumer. SGCC assigns the labels after conducting on-site inspections. The dataset is in tabular form, with rows representing consumers, and columns indicating the daily EC of each consumer from 1 January 2014 to 31 October 2016. Facts and figures regarding the SGCC dataset are mentioned in Table 1. Here, it is important to mention that the dataset contains some incorrect and missing values. Therefore, to handle this issue, data preprocessing is used, as described in Section 4.2.

Table 1. Details about the data.

Description	EC Time Window	Class of Customer	Power Source	Data Resolution	Total Customers	Honest Customers	Thief Customers
Values	1 January 2014 to 31 October 2016	Residential	Utility	Daily data	42,372	38,757	3615

4.2. Data Preprocessing

Data preprocessing is an important step and includes the following steps: removal of missing values and outliers, normalization of data values, feature extraction or selection, and handling the class imbalance problem.

4.2.1. Handling the Missing Values

The SGCC dataset contains missing values and non-numeric values, indicated by 'NAN'. These values occur for many reasons, such as improper operation of smart meters, human typos, data storage problems, and distribution line faults. If the data contain missing values, ML and DL methods do not produce good results. If the records with missing values are removed, it may also take away important information which creates the problem of underfitting. The missing values are tackled with linear imputation to avoid the problem of underfitting. The mathematical equations are given below.

$$f(z_i) = \begin{cases} \frac{z_{i,j-1} + z_{i,j+1}}{2}, & z_{i,j} = NaN, z_{i,j\pm1} \neq NaN, \\ 0, & z_{i,j-1} = NaN \text{ or } z_{i,j+1} = NaN, \\ z_{i,j}, & z_{i,j} \neq NaN. \end{cases} \quad (1)$$

In Equation (1), z_i denotes the EC of consumer i on the current day, and z_{i-1} and z_{i+1} show the EC of the previous day and the next day, respectively.

4.2.2. Removing the Outliers

Some outliers are also found in the data. In the preprocessing of the data, one of the most important steps is to remove or treat the outliers. In the literature, experimental results show the sensitivity of the ML and DL models to splitting data and generating false results. To treat the outliers, the three-sigma rule (TSR) is used in this study. The mathematical equation of the TSR is given below.

$$f(z_i) = (z_i) * \sigma(z_i) \quad if\ z_{i,j} > \mu(z_i) + 3 * \sigma(z) \quad otherwise\ f(z_i) = z_i \tag{2}$$

In Equation (2), z_i shows the EC history of a consumer i, $\mu(z_i)$ represents the averaging of EC, and $\sigma(z_i)$ denotes the standard deviation.

4.2.3. Normalizing the Data Values

After performing the above steps, normalization of the data is done by a min–max method. The reason for this is that ML and DL do not work well on diverse data. The mathematical equation is given below.

$$z_{i,j} = \frac{z_{i,j} - min(Z_i)}{max(Z_i) - min(Z_i)} \tag{3}$$

In Equation (3) $min(Z_i)$, represents the minimum EC, while $max(Z_i)$ denotes the maximum EC of consumer i.

Algorithm 1 shows the data pre-processing phase, which contains following steps: handling the missing values, removing the outliers, and normalizing the data values.

Algorithm 1: Data pre-processing phase.

Data: EC data: Z
$X = (z_{i,j}, y_i), (z_{i+1,j}, y_{i+1}), ..., (z_{m,n}, y_m)$
m = number of records, n = number of features
Variables: min_i = minimum consumption, max_i = maximum consumption, $\overline{z}_i$ = mean consumption, σ_i = standard deviation,
for $i \leftarrow m$ **do**
  for $j \leftarrow n$ **do**
    Handle the missing data:
    if $z_{i,j-1}$ && $z_{i,j+1} \neq NaN$ && $z_{i,j} == NaN$ **then**
      $z_{i,j} = (z_{i,j-1} + z_{i,j+1})/2$
    end
    if $z_{i,j-1} \parallel z_{i,j+1} == NaN$ **then**
      $z_{i,j} = 0$
    end
    Remove anomalies:
    if $z_{i,j} > \overline{z}_i + 3\sigma_i$ **then**
      $z_{i,j} = \overline{z}_i + 3\sigma_i$
    end
    Data normalization through min–max method:
    $z_{i,j} = \frac{z_{i,j} - min_i}{max_i - min_i}$
  end
end
Result: $Z_{normalized} = Z$

4.2.4. Class Imbalance Problem

The problem of class imbalance or uneven distribution of class samples is a severe issue in ETD, where there are more samples of one class than other classes. When ML or DL are trained on an imbalanced dataset, they provide biased results with high FPRs. As mentioned in the literature, power generation companies cannot tolerate high FPRs because they have limited resources for on-site inspections. Two approaches are generally used in the literature to deal with class imbalance problems: undersampling and oversampling. In the former, replicates of the minority class are generated, while in the latter, samples are eliminated to balance the classes. However, both techniques have the following drawbacks: overfitting, duplication of existing data, and loss of information. In this paper, a hybrid sampling approach based on adaptive synthetic sampling (ADASYN) and TomekLinks is proposed. The former uses oversampling while the latter uses undersampling to solve the problem of class imbalance. The proposed hybrid approach solves the problems of undersampling, oversampling, and duplication of data. A detailed description of ADASYN and TomekLinks can be found below.

ADASYN (Adaptive Synthetic):

To solve underfitting, ADASYN is employed to generate minority class samples, which are harder to learn. The overall working mechanism of that sampling approach is elaborated below.

- The ratio of the minority to the majority class is calculated using the below equation:

$$d = \frac{m_{min}}{m_{maj}} \tag{4}$$

  where m_{min} is the total number of minority class samples, and m_{maj} is the number of majority class samples in the dataset.
- The ratio of how many samples will be generated is decided using the following equation:

$$G = (m_{maj} - m_{min})\beta \tag{5}$$

  where G is the total number of minority class samples that will be generated to handle undersampling; β is a random number whose value is between 1 and 0, with 0 indicating that no samples of the minority class will be generated, while 1 shows that minority samples will be generated until both classes have an equal number of samples, $\beta = (0, 1)$.
- In this step, the number of majority class samples near minority class samples are calculated using k-nearest neighbors. After that, each minority class sample is associated with a different number of neighbors that belong to the majority class.

$$r_j = \frac{majority}{k} \tag{6}$$

  Here, r_j shows the dominance of the majority class samples over each minority class sample. A higher r_j shows that it is difficult for ML and DL models to learn/remember the patterns of minority class samples. Thus, a greater number of samples are created for minority class samples that are surrounded by large/maximum numbers of majority class samples. This phenomena gives an adaptive nature to ADASYN.
- To normalize the r_j values, we use

$$r_j = \frac{r_j}{\sum r_j} \qquad \sum r_j = 1 \tag{7}$$

- For minority class samples, we compute the amount of synthetic samples with

$$G_j = Gr_j \tag{8}$$

- In the last step, Algorithm 1 selects the minority class samples from training data and generates new samples. If training data contain m number of minority class samples, then new samples are created using the following equation.

$$s_j = x_j + (x_j - x_{random}) * \lambda, j = 1 \ldots m. \quad (9)$$

In the above equation, λ is a random number between 1 and 0, $_j$ is the newly generated sample, x_j is a first sample of training data, and x_{random} is a randomly selected sample from the training data.

TomekLink:

TomekLink is used for undersampling class imbalance problems. It is a modification of Condensed Nearest Neighbor ((CNN), not to be confused with Convolutional Neural Network). It uses the following rules to select pairs of observations (e.g., X and Y) that satisfy the properties listed below:

- The observation that X's nearest neighbor is Y (and vice versa);
- The observation that X and Y belong to different classes: either the minority class or the majority class.

Mathematically, this is expressed as (X_{min} and X_{maj}), representing the Euclidean distance between X_{min} and X_{maj}, where X_{min} and X_{maj} belong to the minority and majority classes, respectively. If there is no sample X_k that satisfies the following conditions:

$$d(X_{min}, X_k) < d(X_{min}, X_{maj}) \quad (10)$$

$$d(X_{maj}, X_k) < d(X_{min}, X_{maj}) \quad (11)$$

The pair (X_{min}, X_{maj}) are TomekLink samples, which removes noise and duplicated values from data. Consequently, ML and DL models learn diverse patterns from data and do not get stuck in underfitting.

5. Proposed Model

In [5], a combined MLP and CNN model is proposed, which proves that the hybrid model outperforms standalone models of ML and DL. In [22], the authors present CNNs with LSTMs. GRUs and LSTMs utilize different approaches toward gating information to prevent the vanishing gradient problem. RNNs have two variants: GRU and LSTM. The vanishing gradient problem is solved by the author of [31] by comparing the performance of GRU and LSTM with an RNN model using different sequential datasets. Extensive experimentation are performed by Ding et al. on 10,000 LSTM and RNN architectures [32]. The final results advocate that GRU outperform as compared to all contemporary models. For the above reasons, in this research paper a hybrid DL model is presented that combines the advantages of both GRU and CNN models. The GRU extracts the time-related patterns, while the CNN retrieves abstract or latent pattern data. The HGC model consists of the following parts/modules: GRU, CNN, and Hybrid. One-dimensional data are fed as input to the GRU module, while 2D data are fed as input to the CNN to learn abstract features. The hybrid module takes the extracted features from both modules as input and combines them to discriminate between malicious and normal patterns. From the literature, hybrid models work well because they allow combined training and testing of both DL models. In the following, the individual modules are explained in more detail.

5.1. Gated Recurrent Unit (GRU)

GRU is an enhanced form of a recurrent neural network (RNN). One of the main problems in RNNs is the vanishing gradient problem, which stops the learning process and pushes the sequential DL models into local optima. To solve the prior problem, GRU model was introduced. GRU structure consists of an update gate and a reset gate that affect the learning of temporal patterns from EC data. Basically, the information to be passed

to the next layers or units is determined by the update gate. Otherwise, the amount of information from the past that should be forgotten is determined by the reset gate. This information is not important for future decisions. The GRU layers are trained on past data, learn and remember the important information, and remove the redundant values that are not important for distinguishing between malicious and normal patterns. These GRU layers are able to retrieve time-related patterns from EC data. The equations of the update and reset gates are given below.

$$UG_t = \sigma(U_{ug}, [hdn_{t-1}, Z_t]), \tag{12}$$

$$RG_t = \sigma(U_{rg}, [hdn_{t-1}, Z_t]), \tag{13}$$

$$\hat{hdn_t} = \tanh(U, [r_t * hdn_{t-1}, Z_t]), \tag{14}$$

$$hdn_t = (1 - UG_t) * hdn_{t-1} + UG_t * \hat{h}_t. \tag{15}$$

$$Dense_{GRU} = Flaten(hdn_t * W_{GRU} + b_{GRU}) \tag{16}$$

where Z_t and hdn_{t-1} show the input value and hidden layer value of the previous time step, respectively, UG_t indicates the update gate, RG_t shows the reset gate, U_{ug} and $U_{rg}r$ are weights of the update and reset gates, respectively. $Dense_{GRU}$ layers are used to merge extracted features of GRU and CNN models to enhance the prediction accuracy. The hyperparameter settings for GRU are mention in Table 2.

Algorithm 2 describes the working mechanism of the proposed hybrid DL model containing a GRU, a CNN, and fully connected layers.

Algorithm 2: Working of HGC model.

Data: EC data: $Z_{Balance}$
Data in 1D format:
$Z_{1D} = z_{i,j}, z_{i,j+1}, z_{i,j+2}, ..., z_{l,m}$
$l = 42372,\ m = 1034$
Convert data to 2D format
$Z_{2D} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,k} \\ \vdots & \ddots & \vdots \\ x_{j,1} & \cdots & x_{m,k} \end{bmatrix}$
Pass Z_{1D} data to GRU model
for $i <$ *Epoch* **do**
  $r_t = \sigma(U_{rg}, [hdn_{t-1}, x_t])$
  $\hat{hdn_t} = \tanh(U, [r_t * hdn_{t-1}, x_t])$
  $hdn_t = (1 - z_t) * hdn_{t-1} + z_t * \hat{hdn_t}$
  $Dense_{GRU} = relu(U \cdot hdn_t, b])$
  $Fl_{GRU} = flatten(Dense_{GRU})$
  $Z_{2D}[u, v] = (Z_{2D})[m, v] = \sum_j \sum_k f[j, k] Z_{2D}[m - j, v - k]$
  $u, v \Rightarrow$ dimension of output matrix
  $Fl_{CNN} = flatten(Z_{2D})$
  $h_{HGC} = (W_{HGC} \cdot [Fl_{CNN}, Fl_{GRU}] + b)$
  $Dense_{layer} = [U \cdot h_{HGC} + b]$
  $b \Rightarrow$ bias term, $U \Rightarrow$ weight
  $Y_{NTL} = \sigma(Dense_{layer})$
  $Loss(Y_{NTL}, Y) = -\sum_{i=i}^{42372} (Y_i \cdot \log Y_{NTL_i})$
  $Y_{NTL} = Predicted, Y = Actual$
  Reduce the loss value
end
Result: Y_{NTL}

Table 2. GRU hyperparameter settings.

Model	GRU Layers	Activation Function	Dropout Rate	Kernel Initializer	Hyperparameter	Epochs
GRU	40	Sigmoid	0.4	he_{normal}	Optimal values	15

5.2. Convolutional Neural Network (CNN)

The CNN algorithm belongs to the group of DL models. It is mainly used in the recognition of images and videos. It is an extended version of the MLP. It takes images as input, learns important features using a weight-learning mechanism, and develops a relationship between learned features and labels. Technically, CNN design consists of a number of convolution layers with filters (kernels), pooling layers, then one or more fully connected (FC) layers; it applies a softmax function to classify an object with probabilistic values between 0 and 1. Each layer has its own functionality that extracts abstract or latent features that cannot be detected by the human eye. In this study, a CNN model is used to extract latent patterns from data provided by electric utilities. The extracted features are fed into the hybrid layer to make final decisions about malicious and normal consumers. The final hidden layer of the CNN model is shown below.

$$Dense_{CNN} = Flaten(X * W_{CNN} + b_{CNN}) \tag{17}$$

where W_{CNN} and b_{CNN} represent the weight and bias values, respectively, of hidden CNN layers and the feature matrix by X. The hyperparameter settings for CNN are explained in Table 3.

Table 3. CNN hyperparameter settings.

Model	Filters	Strides	Padding	Activation	Batch Size	Epochs	Time
CNN	32	1	Same	ReLu	64	15	202 s

5.3. Hybrid Module

The GRU model learns temporal patterns from 1D data, while CNN extracts the patterns, which are viewed through the human eye from 2D data. The extracted features of both models are concatenated using Keras API and then passed to a hybrid layer that decides whether there is an anomaly in the EC data; h_{HGC} is the last hidden layer of the hybrid module. Its output is passed to the sigmoid function to give a final decision about malicious and normal consumers.

$$h_{HGC} = (W_{HGC}[Dense_{CNN} + Dense_{GRU}] + b_{HGC}), \quad Y_{NTL} = \sigma(h_{HGC}) \tag{18}$$

where W_{HGC} and b_{HGC} represent the weight and bias values of the hybrid layer, and σ denotes a sigmoid function. The settings of hyperparameter for HGC are mention in Table 4 and the pictorial representation of the proposed framework is given in Figure 1.

Table 4. HGC parameter settings.

Model	Layers	Dense	Batch Size	Epochs	Optimize	Time (s)	Dropout	Activation	Kernal Initializer	Pool Size
GRU	40	20	64	10	ADAM	1704	0.4	-	he_{normal}	-
CNN	32	20	64	10	ADAM	1704	-	ReLu	-	2×2

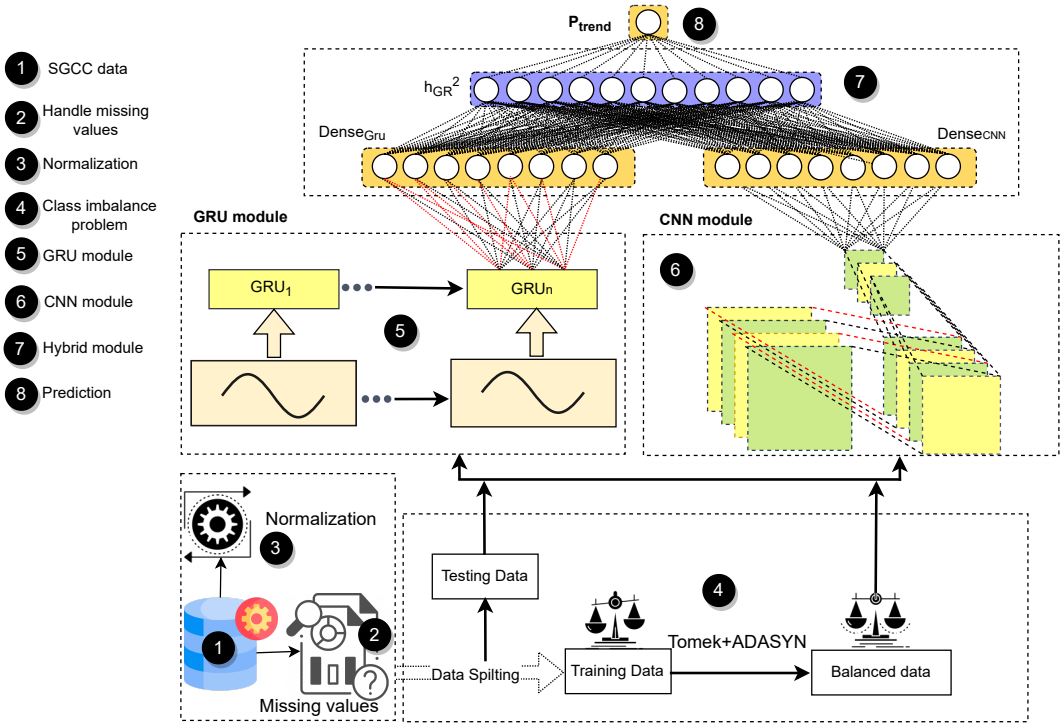

Figure 1. Proposed system model.

6. Experimental Setting and Analysis

In this section, we analyze the performance of the proposed model on the SGCC dataset using various performance measures. We also compare the results obtained with the proposed model to those of benchmark models.

6.1. Performance Measures

Uneven distribution of class samples is a critical problem in ETD, where the number of samples of the normal class is higher than that of the malignant class. When an ML or DL model is trained on this type of data, it attracts majority class samples and ignores minority class samples, producing false results/alarms. The literature indicates that electric utilities cannot tolerate false alarms due to limited resources for on-site testing. Although the training dataset is balanced with the proposed sampling technique, the test data are unbalanced. Therefore, appropriate performance measures are needed to evaluate the performance of the benchmark and proposed models. In this paper, the performance measures used are accuracy, F1 score, recall, ROC-AUC, and PR-AUC. To calculate the above measures, we use a confusion matrix: a confusion table that contains true negative (TN), true positive (TP), false negative (FN), and false positive (FP) results.

6.1.1. Accuracy

Accuracy is the ratio between the number of correct predictions and the total number of records in the dataset.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{19}$$

where TN and TP are the sums of total number of true negatives and true positives, respectively, and TN, TP, FN, and FP are the sums of true negatives, true positives, false negatives, and false positives, respectively.

6.1.2. Recall

Recall is determined by dividing the correctly predicted positive records by the total number of positive records. The equation of recall is given below, as described in [33]:

$$Recall = \frac{TP}{FN + TP} \tag{20}$$

where *FN* is the number of dishonest consumers predicted by the model as honest consumers.

6.1.3. F1-Score

The F1-score is also a good performance measure for imbalanced datasets. When ML/DL models have a high F1-score, they are considered good for predictions in real-world scenarios. The equation for the F1-score is given below, as described in [34,35]

$$F1 - Score = \frac{2 * precision * recall}{precision + recall} \tag{21}$$

To calculate the precision, the number of true positives divided by the sum of false positives and true positives, as mentioned in [33].

The ROC curve is obtained by plotting recall and FPR on the y-axis and x-axis, respectively. It is a good measure for imbalanced datasets because it is not skewed toward the majority class. Its value ranges from 0 to 1. However, ROC only considers the recall/true positive rate, so it focuses on positive records and ignores the negative ones. The PR curve is another important measure that considers recall and precision simultaneously and gives equal importance to twain classes.

6.2. Implementation Environment

The proposed and benchmark models are implemented using Google Colaboratory [36], which provides distributed computing power. Their performance is studied using the SGCC dataset collected from the largest electric utility in China. DL models are implemented using TensorFlow (v2.8.2), while ML models are trained and evaluated using the Scikit library (v1.0.2), and the Keras API is used to develop the hybrid model.

6.3. Proposed Deep Learning Model Performance Analysis

In this section, we analyze the performance of the proposed model using accuracy and loss curves for training and testing data. Figure 2 shows the performance of the model on training and test data using accuracy curves. Both curves move side-by-side with a small difference, indicating that the proposed model does not suffer from overfitting. However, after the fourth epoch, the test accuracy starts to decrease, which means that the model suffers from overfitting. Thus, if more than four epochs are trained, the performance of the model decreases. To improve the model's performance in the future, meta-heuristic algorithms will be used to help select the optimal parameters for deep and machine learning to avoid overfitting. It is very complex and time-consuming to select these parameters manually.

Figure 3 also shows the same phenomena using loss curves on training and testing data. The value of loss can be decreased with more epochs.

However, there is a high probability that the model encounters overfitting, which affects generalization. In addition, the proposed model consists of GRU, CNN, and dense layers. The gates like, update and reset in the GRU layer control the information flow through network. These gates remember valuable information and ignore redundant and noisy patterns from the data. CNN layers help the proposed hybrid model learn global/abstract patterns from EC data and reduce the curse of dimensionality, which directly increases the convergence speed. The literature shows that dropout layers simplify the model and prevent overfitting. Finally, the dense layer takes inputs from the GRU and CNN models and passes them to a sigmoid function to distinguish between normal

and malicious samples. For all these reasons, a hybrid model performs better than the individual models.

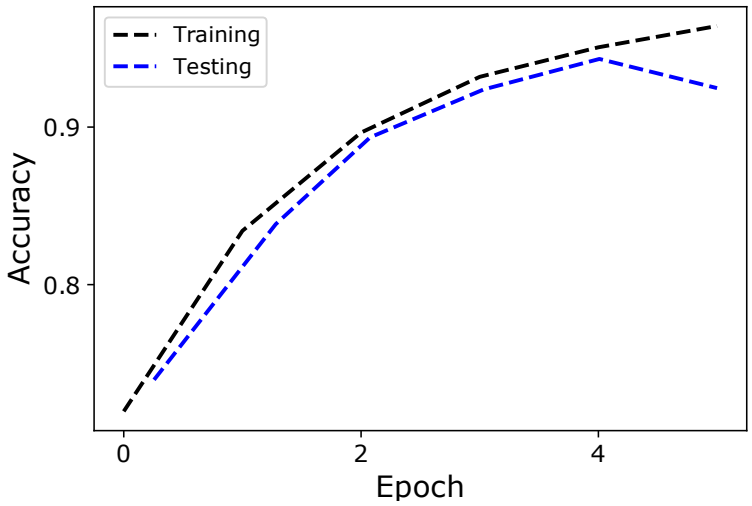

Figure 2. Accuracy curves on training and testing data.

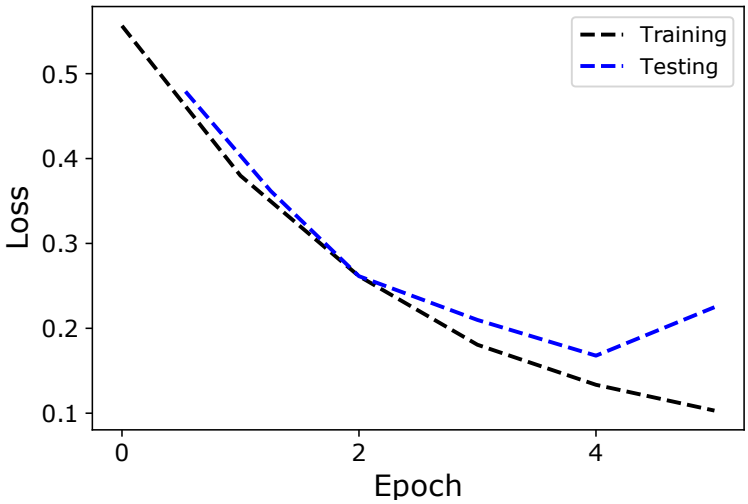

Figure 3. Loss curves on training and testing data.

6.4. Benchmark Models

This section implements various DL and ML models that have previously been proposed in the literature and compares their performance with that of the proposed hybrid model.

6.4.1. Wide and Deep Convolutional Neural Network

In [5], Zheng et al. propose a DL model that is a fusion of CNN and ANN. This is the first study to combine the advantages of both models. The authors feed 2D data to a CNN, while 1D data are fed into an ANN to learn local and global patterns from the SGCC dataset. However, the ANN model does not give good results on 1D data because it is designed for tabular data. In this work, we use the same hyperparameter settings and the same dataset for a fair comparison.

6.4.2. Logistic Regression (LR)

This is a basic supervised learning model used for binary classification. It is also known as a single-layer neural network. It simply contains an input layer whose values are multiplied by weights, and the resulting value is fed into a sigmoid function that

produces either 0 or 1 as input. LR consists of various solvers such as Newton's method and stochastic gradient descent that are used to tune the hyperparameters.

6.4.3. Decision Tree (DT)

DTs are used in both regression and classification tasks. They consist of a root node, edges, and leaf nodes that are used to predict the result. A DT works like the human mind and creates a tree-like structure in which the dataset is divided into many branches based on features. The best attributes/features are selected based on the information gain and Gini index criteria as root nodes. DTs are easy to implement and give good results on smaller datasets. However, for larger datasets there is a risk of overfitting. In addition, a small change in the data leads to poor generalization.

6.4.4. Support Vector Machine (SVM)

SVMs are a supervised learning model used for both regression and classification purposes. They are able to classify linear and nonlinear data by using the power of kernel functions. These kernel functions draw a decision boundary to classify between normal and malicious samples after converting non-linear data into linear patterns. In [7], the authors develop a current theft detector based on consumption patterns using an SVM classifier to draw a decision boundary between benign and stolen samples. From the literature, SVM is well-suited for smaller datasets, as it requires a lot of computational time to draw a decision boundary between normal and malicious patterns for larger datasets. In this work, the RBF kernel is used for the SGCC dataset due to the nonlinearity of the data.

6.4.5. Random Forest (RF)

An ensemble technique called RF is used to solve complex problems by training multiple decision trees on datasets. It has applications in banking, e-commerce, and other fields. RFs control the problem of DF overfitting and increase precision. They give good results with little adjustment of hyperparameters. They also minimize overfitting and increase the precision when the number of DTs is increased during the training period. However, they require a lot of computation time for larger datasets, since multiple DTs are trained on a single dataset, which reduces their effectiveness in real-world problems.

6.4.6. Naive Bayes Classifier

This is a classification method derived from Bayes' theorem. The Naive Bayes (NB) does not consider the linkage between inputted features and targeted column, and uses the probability distribution to distinguish between normal and malicious samples. There are many versions developed depending on the type of dataset. In today's world, there are many applications in various fields such as sentiment analysis, email filtering, recommender systems, spam, and natural language processing. In this work, we use Gaussian NB since the SGCC dataset has continuous features.

7. Experimental Results and Discussions

The performance of the proposed HGC model is compared with the state-of-the-art classifiers. The same datasets with different ratios for training and testing are used for DT, NB, LR, CNN, GRU, RF, SVM, and WDCNN. As discussed earlier, the CNN design consists of a number of convolution layers with filters (kernels) and pooling layers, followed by one or more fully connected (FC) layers, and applies a softmax function to classify an object with probabilistic values between 0 and 1. Each layer has its own functionality and extracts abstract or latent features that cannot be detected by the human eye.

The GRU layers have two important gates; update and reset. These are used to learn necessary patterns and remove unnecessary values. As discussed earlier, the flow of information is controlled by GRU gates to improve the performance of the model. The GRU-extracted features are then combined with the latent or abstract patterns. The proposed HGC model extracts abstract and periodic patterns from EC data using GRU

and CNN hence HGC outperforms as compared to counterparts of it. The combination of optimal features helps the HGC to attain 0.96 PR-AUC and 0.97 ROC-AUC values, which are higher than those of all the above-mentioned classifiers. The performance of proposed model is compared with conventional models using PR and ROC curves in Figures 4 and 5. The proposed hybrid model achieves better results than its counterparts. SVM achieves 0.88 ROC-AUC and 0.85 PR-AUC. We use a linear kernel instead of an RBF kernel to train the SVM model on EC data because the dataset contains a large number of records and features, which increases the model computation time, so it is not suitable for larger datasets.

LR is a conventional ML model that distinguishes between normal and malignant samples using a sigmoid function. It achieves 0.86 and 0.88 for PR-AUC and ROC-AUC, respectively, which is better than SVM, but has lower performance than other models. It has a large number of applications in various fields because it is easy to implement and is suitable for linearly separable datasets, but in the SGCC dataset, malicious and normal samples are not linearly separable. Therefore, LR gives lower performance compared to other models [30].

RF gets 0.76 PR-AUC and 0.75 ROC-AUC, while DT gets 0.80 ROC-AUC and 0.85 PR-AUC on the EC dataset. DT gives better results than RF. DT provides good performance on smaller datasets but has overfitting on larger datasets, and small changes in the data reduce its generalization ability. RF is an ensemble method designed to overcome the overfitting/low generalization of DT. It controls overfitting but has low PR-AUC and ROC-AUC, as seen in Figures 4 and 5, because RF takes the average of all DT prediction results.

In addition, NB is a conventional classifier that classifies between normal and malignant samples using Bayes theorem. It obtains 0.71 and 0.65 PR-AUC and ROC-AUC values, respectively. Unlike other conventional ML and ensemble models, it gives poor results. It assumes that there is an independent relationship between the attributes and the target features.

Moreover, CNN gains 0.96 ROC-AUC and 0.94 PR-AUC values, while GRU gains 0.96 and 0.96 ROC-AUC and PR-AUC values on the EC dataset, which are higher than the PR-AUC and ROC-AUC values of conventional ML models. Technically, a CNN consists of a number of convolution layers with filters (kernels) and pooling layers, followed by one or more fully connected (FC) layers. In addition, the convolutional layer is used to remove redundant, overlapping, and noisy values from the EC data. GRU also gives good results that are in the acceptable range, as it has update and reset gates to help remember periodic patterns. In [5], the authors combine the merits of the ANN and CNN models to develop a hybrid model. Their proposed model achieves a value of 0.96 PR-AUC and 0.97 ROC-AUC. In the literature, the authors demonstrate that the hybrid model performs better than the DL models and the standalone ML model. Therefore, in this research, the Keras API is used to develop a hybrid model. It integrates the advantages of both GRU and CNN models. The former learns the temporal patterns, while the latter derives global and abstract patterns from EC data. The extracted features of both models are merged and passed to a fully linked layer for the classification of theft and normal patterns. The proposed model achieves better results than the standalone DL and the previously proposed hybrid DL models for the above reasons. It achieves 0.987 ROC-AUC values and 0.985 PR-AUC values on EC data, as observed in Tables 5 and 6.

Tables 5 and 6 show the performance analysis of the ML and DL models at 70% and 60% training ratios, respectively. It can be seen that the proposed model maintains its superiority and gives better results at both training ratios. For the DL models, performance increases as the size of the training data increases because DL models are inherently sensitive to the size of the training data. On the other side, the increased or decreased performance of conventional ML models follow the power law [37]. This law states that beyond a certain point, the performance of ML models increases with the increase of the amount of data. After this point, the models face the problem of overfitting, which affects their generalizability. In this work, RF and NB give poor results compared to other

conventional ML models. Although both models perform well on balanced datasets, they show poor performance due to the following limitations.

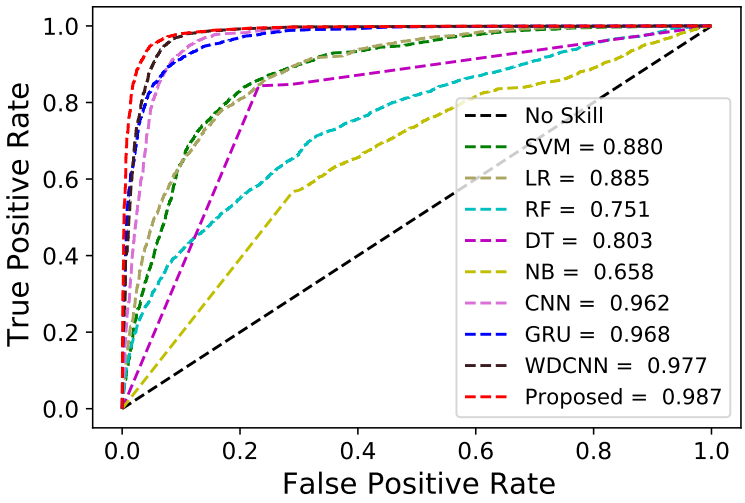

Figure 4. ROC curves of proposed and benchmark models.

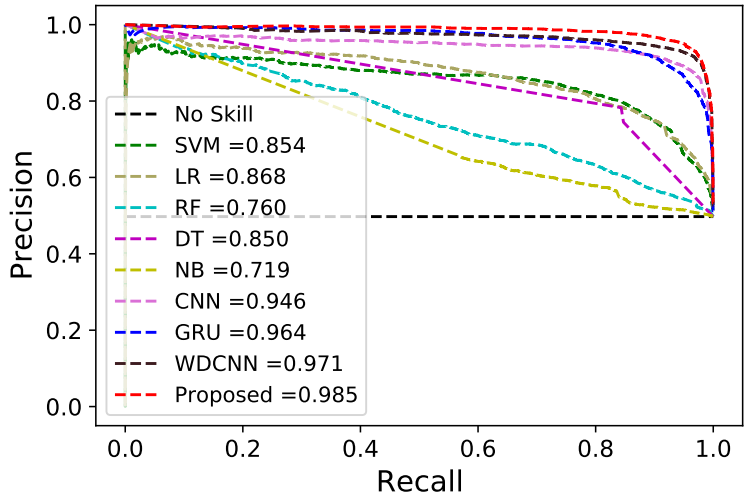

Figure 5. PR curves of proposed and benchmark models.

Table 5. Performance analysis of DL and ML using 70% training data.

ML/DL Models	Accuracy	F1-Score	Recall Score	PR_{AUC}	ROC_{AUC}
LR	0.8040	0.8068	0.714	0.868	0.885
SVM	0.8165	0.8200	0.800	0.854	0.880
RF	0.6912	0.696	0.6128	0.756	0.748
DT	0.8056	0.8118	0.7826	0.850	0.803
NB	0.6261	0.649	0.608	0.719	0.658
CNN	0.914	0.918	0.877	0.946	0.962
GRU	0.9074	0.9080	0.919	0.964	0.968
WDCNN	0.9397	0.9408	0.919	0.971	0.977
HGC	0.9438	0.9452	0.91709	0.985	0.987

Table 6. Performance analysis of DL and ML using 60% training data.

ML/DL Models	Accuracy	F1-Score	Recall Score	PR_{AUC}	ROC_{AUC}
LR	0.804	0.807	0.796	0.868	0.883
SVM	0.811	0.815	0.797	0.855	0.877
RF	0.677	0.680	0.672	0.756	0.748
DT	0.801	0.808	0.781	0.848	0.799
NB	0.619	0.645	0.604	0.715	0.650
CNN	0.916	0.916	0.916	0.955	0.966
GRU	0.925	0.926	0.910	0.968	0.973
WDCNN	0.936	0.938	0.906	0.971	0.779
HGC	0.947	0.948	0.921	0.985	0.987

NB accounts for the independent relationship between features and target variables that does not exist in real EC data, while RF controls for overfitting by the average performance of all DTs. The literature shows that the performance of DL models depends on the size of the training data. Large datasets yield high values for performance measures. ROC analysis of different hybrid models is given in Table 7. In [38], CNN-LSTM and LSTM RUSBoost achieve 0.817 and 0.879 ROC values, respectively, while in [30], MLP–LSTM achieves 0.92 ROC, and HG^2 achieves 0.93 ROC. In our case, our proposed model maintains its superiority and performs better than the above-mentioned hybrid models by achieving 0.98 ROC.

The computation time of the ML and DL models is given in Table 8. NB and LR have a lower computation time in contrast to other ML models because the former only computes the probability distribution of all features and provides the final results, whereas LR is a single-layer neural network that multiplies the inputs with weights and distinguishes between malignant and normal samples. For the above reasons, they require little computational time compared to other ML models.

In ETD, SVM is a well-known classifier. RF requires more training time than DT because it trains multiple DTs on the SGCC dataset and computes the average of multiple estimators. Moreover, the training time of DL models depends on the number of hidden layers, the size of the dataset, the stack size, and the number of neurons in each layer. GRU and CNN are DL models that take 2364 and 202 seconds to train, respectively. GRU requires more training time because it has update and reset gates that extract temporal patterns from SGCC data and save the important information in memory networks, while CNN only retrieves abstract/latent patterns by using convolution functions and max-pooling layers, which is why they have low computation time. Moreover, HGC takes 1704 seconds to train with the SGCC dataset. It has a lower computation time than GRU because it converges in 5 epochs, whereas GRU converges in 15 epochs. In addition, HGC requires more training time than the CNN model because it integrates the benefits of both models. Moreover, at the present time, meta-heuristic techniques are receiving attention from the research community for feature selection and hyperparameter optimization in ML and DL models. Therefore, in this study, BHA, a meta-heuristic technique, is used for feature selection. The literature demonstrates that these techniques have high computational complexity. For this reason, a small portion of the dataset is used to evaluate the ability of BHA for feature selection. The selected data consist of 10,000 records and 30 days of EC values from 42,372 records. BHA takes 3000 seconds to select the optimal combination of features/attributes from the selected EC data, which is more than the time required by all DL models: GRU, CNN, WDCNN, and HGC. The above results show that the computational time of BHA increases as the amount of data increases. Therefore, these types of real-time applications are not suitable for the smart grid. Moreover, the increased dataset size enhances the performance of DL models. Hence, the performance of these models depend on the size of training dataset. In canse of convolution ML models, the

performance is enhanced by following the power law. Their performance stop improving after certain point of training [37].

From the literature, hybrid models work well because they combine training and testing of both DL models and have better generalization capabilities than many other machine and deep learning models. However, HGC maintains dominance over the state-of-the-art DL models and shows better performance on varieties of training ratios over SGCC dataset. Nexus to the above, there is no free lunch. The cost benefit analysis is a trade-off between computational time and accuracy. The proposed algorithm is computationally expensive, but on the other hand, it provides higher accuracy than the other algorithms used for comparison. With more and more computational resources available these days, researchers are focusing on algorithms that provide better efficiency in the face of widespread data.

Table 7. ROC performance analysis of hybrid models.

Hybrid Models	CNN-LSTM	LSTM-RUSBoost	MLP-LSTM	HG^2	Proposed Model
ROC	0.817	0.879	0.92	0.93	0.98

Table 8. Computation time of ML and DL models.

ML/DL Models	SVM	LR	DT	RF	NB	SVM + BHA	CNN	GRU	WDCNN	HGC
Time (s)	1618	4	52	281	1	3000	202	2364	304	1704
Epoch	-	-	-	-	-	-	15	15	15	5

8. Conclusions and Future Work

Electricity theft is an unavoidable issue that causes power losses in both; developed and developing countries. As a result, power utility companies have major disruptions in their operations, leading to loss of revenue. Moreover, electricity loss also causes issues with economic growth and power infrastructure stability. In this study, a combined DL model for NTL detection is presented that incorporates a GRU and a CNN. To remove null and undefined values, EC data are pre-processed by normalization. In addition, uneven distribution of class samples is another problem in ETD that affects the effectiveness of the ML and DL models. In this paper, a hybrid approach is used to address these problems. The performance of the proposed model is evaluated on the SGCC dataset in real-time using various performance metrics and compared with SVM, LR, CNN, GRU, RF, DT, NB, and WDCNN. The model achieves 0.987, 0.985, 0.94, 0.94, and 0.91 ROC-AUC, PR-AUC, accuracy, F1-score, and recall score on the SGCC dataset, respectively. The obtained results are better than those of other ML and DL models. However, despite the proposed model outperforming substitute techniques, it is too sensitive to changes in input data. The presented model will help many industrial applications to identify normal and abnormal samples or records. To improve the model's performance and avoid overfitting, meta-heuristic algorithms help select the optimal parameters for deep and machine learning. It is very complex and time consuming to select these parameters manually.

In the future, meta-heuristic techniques will be used to achieve optimal hyperparameter tuning in DL models.

Author Contributions: Conceptualization, A.K., R.B. and R.A.; Data curation, A.K.; Methodology, R.B., O.A. and R.A.; Project administration, S.A. and A.Y.; Resources, A.Y. and O.A.; Software, A.Y. and O.A.; Supervision, R.B. and S.A.; Validation, S.A.; Visualization, R.A.; Writing—original draft, A.K. and R.B.; Writing—review & editing, R.B., S.A., A.Y., O.A. and R.A. All authors have contributed equally and have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors of this study would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions to improve our work.

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

Nomenclature

ANN	Artificial Neural Network (NN)	z_i	EC of consumer i at current day
ADASYN	Adaptive Synthetic	z_{i-1}	EC of consumer i at previous day
AUC	Area Under the Curve	z_{i+1}	EC of consumer i at next day
BHA	Black Hole Algorithm		
CNN	Convolutional Neural Network	$\mu(z_i)$	Represents average E
DT-SVM	Decision Tree-SVM	$min(Z_i)$	Minimum EC
DL	Deep Learning	$max(Z_i)$	Maximum EC
DE	Differential Evolution	m_{min}	Total number of minority class
DT	Decision Tree	m_{max}	Total number of majority class
DNN	Deep Neural Network	G	Total number of minority data to be generated
EC	Electricity Consumption	ETD	Electricity Theft Detection
FP	False Positive	FN	False Negative
FPR	False Positive Rate	GRU	Gated Recurrent Unit
HGC	Hybrid GRU–CNN	LSTM	Long Short-Term Memory
KNN	K-Nearest Neighbor	β	Ratio of minority: majority data desired after ADASYN
LR	Linear Regression	MLP	Multi-Layer Perceptron
NTL	Non-Technical Loss	RNN	Recurrent Neural Network
PR-AUC	Precision–Recall Area Under Curve	ROC-AUC	Receiver Operating Characteristic Area Under Curve
RBF	Radial Basis Function	RF	Random Forest
SGCC	State Grid Corporation of China	SVM	Support Vector Machine
TP	True Positive	TN	True Negative
TL	Technical Loss	TSR	Three Sigma Rule
WADCNN	Wide And Deep Convolution NN	λ	Number between 0–1

References

1. Leon, C.; Biscarri, F.; Monedero, I.; Guerrero, J.; Biscarri, J.; Millan, R. Variability and Trend-Based Generalized Rule Induction Model to NTL Detection in Power Companies. *IEEE Trans. Power Syst.* **2011**, *26*, 1798–1807. [CrossRef]
2. Glauner, P.; Meira, J.; Valtchev, P.; State, R.; Bettinger, F. The Challenge of Non-Technical Loss Detection Using Artificial Intelligence: A Survey. *Int. J. Comput. Intell. Syst.* **2017**, *10*, 760. [CrossRef]
3. McLaughlin, S.; Holbert, B.; Fawaz, A.; Berthier, R.; Zonouz, S. A Multi-Sensor Energy Theft Detection Framework for Advanced Metering Infrastructures. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 1319–1330. [CrossRef]
4. David, N. The Effects of Energy Theft on Climate Change and Its Possible Prevention Using Smart Meters: Case Study Nigeria. *Int. J. Sci. Eng. Res.* **2018**, *9*. 1775.
5. Zheng, Z.; Yang, Y.; Niu, X.; Dai, H.; Zhou, Y. Wide and Deep Convolutional Neural Networks for Electricity-Theft Detection to Secure Smart Grids. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1606–1615. [CrossRef]
6. Aryanezhad, M. A novel approach to detection and prevention of electricity pilferage over power distribution network. *Int. J. Electr. Power Energy Syst.* **2019**, *111*, 191–200. [CrossRef]
7. Jokar, P.; Arianpoo, N.; Leung, V. Electricity Theft Detection in AMI Using Customers' Consumption Patterns. *IEEE Trans. Smart Grid* **2016**, *7*, 216–226. [CrossRef]
8. Lo C.; Ansari, N. Consumer: A Novel Hybrid Intrusion Detection System for Distribution Networks in Smart Grid. *IEEE Trans. Emerg. Top. Comput.* **2013**, *1*, 33–44. [CrossRef]
9. Xiao, Z.; Xiao, Y.; Du, D. Non-repudiation in neighborhood area networks for smart grid. *IEEE Commun. Mag.* **2013**, *51*, 18–26. [CrossRef]
10. Khoo, B.; Cheng, Y. Using RFID for anti-theft in a Chinese electrical supply company: A cost-benefit analysis. In Proceedings of the IEEE Wireless Telecommunications Symposium, New York, NY, USA, 13–15 April 2011; pp. 1–6.
11. Angelos, E.; Saavedra, O.; Cortés, O.; de Souza, A. Detection and Identification of Abnormalities in Customer Consumptions in Power Distribution Systems. *IEEE Trans. Power Deliv.* **2011**, *26*, 2436–2442. [CrossRef]
12. Depuru, S.; Wang, L.; Devabhaktuni, V.; Nelapati, P. A hybrid neural network model and encoding technique for enhanced classification of energy consumption data. In Proceedings of the IEEE Power and Energy Society General Meeting, San Diego, CA, USA, 24–29 July 2011; pp. 1–8.

13. Depuru, S.; Wang, L.; Devabhaktuni, V.; Green, R. High performance computing for detection of electricity theft. *Int. J. Electr. Power Energy Syst.* **2013**, *47*, 21–30.
14. Jiang, H.; Wang, K.; Wang, Y.; Gao, M.; Zhang, Y. Energy big data: A survey. *IEEE Access* **2016**, *4*, 3844–3861. [CrossRef]
15. Batalla-Bejerano, J.; Trujillo-Baute, E.; Villa-Arrieta, M. Smart meters and consumer behaviour: Insights from the empirical literature. *Energy Policy* **2020**, *144*, 111610. [CrossRef]
16. 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]
17. Ramos, C.C.O.; 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]
18. Fenza, G.; Gallo, M.; Loia, V. Drift-aware methodology for anomaly detection in smart grid. *IEEE Access* **2019**, *7*, 9645–9657. [CrossRef]
19. 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*, 4136874. [CrossRef]
20. Buzau, M.M.; Tejedor-Aguilera, J.; Cruz-Romero, P.; Gómez-Expósito, A. Hybrid deep neural networks for detection of non-technical losses in electricity smart meters. *IEEE Trans. Power Syst.* **2019**, *35*, 1254–1263.
21. Rouzbahani, H.M.; Karimipour, H.; Lei, L. An Ensemble Deep Convolutional Neural Network Model for Electricity Theft Detection in Smart Grids. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 3637–3642.
22. Ghori, K.M.; Abbasi, R.A.; Awais, M.; Imran, M.; Ullah, A.; Szathmary, L. Performance analysis of different types of machine learning classifiers for non-technical loss detection. *IEEE Access* **2019**, *8*, 16033–16048. [CrossRef]
23. Buzau, M.; Tejedor-Aguilera, J.; Cruz-Romero, P.; Gomez-Exposito, A. Detection of Non-Technical Losses Using Smart Meter Data and Supervised Learning. *IEEE Trans. Smart Grid* **2019**, *10*, 2661–2670.
24. Kong, X.; Zhao, X.; Liu, C.; Li, Q.; Dong, D.; Li, Y. Electricity theft detection in low-voltage stations based on similarity measure and DT-KSVM. *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106544. [CrossRef]
25. Hasan, M.; Toma, R.; Nahid, A.; Islam, M.; Kim, J. Electricity Theft Detection in Smart Grid Systems: A CNN-LSTM Based Approach. *Energies* **2019**, *12*, 3310. [CrossRef]
26. Ismail, M.; Shaaban, M.; Naidu, M.; Serpedin, E. Deep Learning Detection of Electricity Theft Cyber-Attacks in Renewable Distributed Generation. *IEEE Trans. Smart Grid* **2020**, *11*, 3428–3437. [CrossRef]
27. Maamar, A.; Benahmed, K. A Hybrid Model for Anomalies Detection in AMI System Combining K-means Clustering and Deep Neural Network. *Comput. Mater. Contin.* **2019**, *60*, 15–39. [CrossRef]
28. Li, W.; Logenthiran, T.; Phan, V.; Woo, W. A Novel Smart Energy Theft System (SETS) for IoT-Based Smart Home. *IEEE Internet Things J.* **2019**, *6*, 5531–5539. [CrossRef]
29. Manoharan, H.; Teekaraman, Y.; Kirpichnikova, I.; Kuppusamy, R.; Nikolovski, S.; Baghaee, H.R. Smart grid monitoring by wireless sensors using binary logistic regression. *Energies* **2020**, 15, 3974. [CrossRef]
30. Shehzad, F.; Javaid, N.; Almogren, A.; Ahmed, A.; Gulfam, S.M.; Radwan, A. A Robust Hybrid Deep Learning Model for Detection of Non-Technical Losses to Secure Smart Grids. *IEEE Access* **2021**, *9*, 128663–128678. [CrossRef]
31. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
32. 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**, *79*, 106458. [CrossRef]
33. Hand, D.; Christen, P. A note on using the F-measure for evaluating record linkage algorithms. *Stat. Comput.* **2018**, *28*, 539–547. [CrossRef]
34. Gu, Y.; Cheng, L.; Chang, Z. Classification of Imbalanced Data Based on MTS-CBPSO Method: A Case Study of Financial Distress Prediction. *J. Inf. Process. Syst.* **2019**, *15*, 682–693.
35. Douzas, G.; Bacao, F.; Fonseca, J.; Khudinyan, M. Imbalanced Learning in Land Cover Classification: Improving Minority Classes' Prediction Accuracy Using the Geometric SMOTE Algorithm. *Remote Sens.* **2019**, *11*, 3040. [CrossRef]
36. Bisong, E. *Building Machine Learning and Deep Learning Models on Google Cloud Platform*; Springer: Berkeley, CA, USA, 2019.
37. Sun, C.; Shrivastava, A.; Singh, S.; Gupta, A. Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
38. Adil, M.; Javaid, N.; Qasim, U.; Ullah, I.; Shafiq, M.; Choi, J. LSTM and Bat-Based RUSBoost Approach for Electricity Theft Detection. *Appl. Sci.* **2020**, *10*, 4378. [CrossRef]