International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Energy Theft Detection with Determine Date Theft Period for State Grid Corporation of China Dataset

Mali H. Alameady[1] Loay E. George[2] Salah Albermany[3]

Submitted: 14/10/2022 **Revised**: 16/12/2022 **Accepted**: 03/01/2023

Abstract: Electricity theft is a major concern for electric power distribution companies. The data set from the State Grid Corporation of China (SGCC) is preprocessing; first, order dataset by date, second, remove the empty record from the dataset, third, missing values by linear interpolation, and finally, imbalanced data handling technique. Then find feature extraction including monthly average, slope, moment and standard deviation, Variance, Peak to Peak, Energy Entropy, Skewness, Crest Factor, Total harmonic distortion, Log Energy, and Kurtosis for all months with gather. After finding features of the dataset, Deep Convolution Neural Network (DCNN) applied DCNN with A map is classified using a convolution layer to extract features, followed by a softmax layer. DCNN is used for data classification in energy theft or non-theft. Finally, the calculated accuracy achieved 100%, which is quite promising in comparison to other reported categorization schemes. The number and date of the theft were then calculated for each of the records in which the theft occurred.

Keywords: *Electricity Theft Detection, SGCC dataset, Statistical Features, Deep Learning, DCNN, Date of Theft.*

1. Introduction

Electricity loss is the difference between the amount of energy injected and the amount of energy supplied to consumers. Electricity losses in a power system are primarily due to the generation, transmission, and distribution of electrical energy [1]. There is much research using the SGCC dataset for detecting electricity theft, such as Bohani Farah Aqilah's (2021), Many supervised learning approaches, such as decision trees (DT), artificial neural networks (ANN), deep artificial neural networks (DANN), and AdaBoost, are compared in terms of accuracy, recall, precision, AUC, and F1 scores. The data used in this investigation were given by the State Grid Corporation of China (SGCC) [2]. Noor Ibrahim (2021) The architecture of the smart grid (SG) generates data that includes each consumer's power use. With this knowledge, machine learning and deep learning algorithms could be able to recognize power thieves. It is possible to identify automated power theft using a CNN model. To categorize and recognize electricity theft, this study proposes experimenting with different configurations of the sequential model (SM). Two layers of 128 nodes and 64 nodes have been found to provide the most efficient performance. The accuracy got up to 0.92 [3].

In [4], proposes a smart grid energy theft detecting technology that preserves privacy. CNN is used to find irregularities in long-term metering data. Paillier algorithm protects your energy usage's anonymity. Sensitive energy consumption data is safely communicated with little loss. Their security study shows that their system protects data and authenticates users. Their revised CNN model can detect deviant behaviors with 92.67 percent accuracy.

In [5], describes a CNN-LSTM electrical theft detection system. CNN automatically extracts and classifies features. Since smart grid power consumption data is a time series, they developed a CNN-LSTM model for data categorization. When a dataset has gaps, a revolutionary data pre-processing method calculates missing instances. Very few electrical theft victims may have hampered the model's accuracy. Synthetic data was used to remedy this inequality. The results show that the suggested technique can consistently separate paying and stealing electricity customers.

In [6], the authors described an energy theft detection technique based on power providers' power consumption data collection systems to identify energy theft at the edge. The steps are as follows. Using K-means and neural network parameters, the centralized data center decomposes large amounts of data into little data for feature extraction. For more precise features, they create DWMCNN, which can extract day, week, and month features. In the edge data center, RF classifies collected characteristics. Clustering speeds up edge computing-distributed processing, according to tests. The feature extractor is convergent. It's more precise and less computationally demanding than previous multiple-classifier techniques, making it perfect for edge data centers.

In [7], suggested Ensemble-based deep learning can identify erroneous readings in real-time. A sliding window of readings is used to train deep-learning models. The best-performing model is used to train other models on a range of false reading ratios; these models are employed in their ensemble-based detection approach. Extensive testing shows that, compared to the typical daily and weekly detection processes, which require 144 and 1,008 measurements, their detector can detect

[1]*Computer Science Department, Faculty of Computer Science and Mathematics,University of Kufa, Najaf, Iraq*
maali.alameedi@uokufa.edu.iq
[2]*University of Information Technology and Communication, Baghdad, Iraq*
loayedwar57@yahoo.com
[3]*Computer Science Department,Faculty of Computer Science and Mathematics,University of Kufa, Najaf, Iraq*
salah.albermany@uokufa.edu.iq

erroneous readings after only a few (around 15).

In [8], proposes a three-part ETD structure. The first module handles power anomalies, outliers, and non-standard data. Second-module class balancing uses a hybrid method. The third module uses an artificial neural network (ICANN) to anticipate electricity theft. A conventional neural network (ANN) may be enhanced by changing hyper-parameters, regularization, and skip connections to classify smart meters. The additional study improves the final classification's generalization and function-fitting. Numerical studies reveal that the ETD model outperforms machine learning and deep learning on real-world energy datasets. This model is industrial.

In [9], introduces a novel hybrid DL model for smart grid power theft detection. AlexNet handles dimensionality while adaptive boosting classifies energy thieves and normal consumers (AdaBoost). Near miss sampling addresses class imbalance. AdaBoost and AlexNet hyper-parameters are also optimized with bees. The hybrid approach uses smart meter data. Simulations demonstrate the hybrid model is the most accurate classifier. Their recommended model achieves 88 percent accuracy, precision, recall, F1-score, Matthew correlation coefficient, and receiver AUC, respectively.

In [10], introduces Data obtained by Smart Grid Infrastructure (SGCC) including electricity usage per consumer. By utilizing a 16-layer CNN model with high data processing and extracting a set of statistical characteristics, algorithms based on machine learning and deep learning may be able to identify electric force thieves using this data. The result is quite high. To classify and identify electricity theft.

2. The SGCC Dataset

This section covers the attributes of the dataset, where the electricity theft data was obtained from the State Grid Corporation of China (http://www.sgcc.com.cn/) [11]. Data set for 34 months. Smart meters or user sensors collect electricity usage data. The data network collects data. In this case, the smart meter, sensor, or data transmission server may fail to store and therefore the electricity consumption statistics will contain missing or incorrect data. This data set contains missing values. If the missing duplicates are eliminated, the data set decreases, making analysis difficult. SGCC provides real-time energy usage data for the investigation. 42,372 rows and 1,035 columns make up this collection. The customer ID is in the first column, a prediction indication called "Tag" is in the second column, and the day's columns begin in the third column (1035). The collection's non-numeric metadata categories include values, characters, and numbers that are missing or wrong. The two-year and ten-month energy usage for each user is represented by missing or erroneous numbers and statistics. The flag column's information (zero and one) identifies the different customer types (normal or thief). The total number of zeros in the "Flag" column represents the average amount of energy consumed, and it is 1. (38757). While "the flag" only mentions one thief, there are a total of (3615). The number (42372) refers to the statistics of total energy consumers' usage for a period of 1,035 days (from January 1, 2014, to October 31, 2016).

Table 1: Metadata information of the original electricity theft dataset.

Description	Value
Time window of data collection	1 January 2014–31 October 2016
Total number of consumers	42 372
Number of normal users	38 757 approx. 91.5%
Number of aberrant users or electricity thieves	3 615 approx. 8.55%
Missing data cases approx.	25.6393%

Smart meters or sensors collect electricity usage data and the network collects data. In this case, the smart meter, sensor, or data transmission server may fail to store it. Consequently, the electricity consumption statistics will contain missing or incorrect data. This data set contains missing values. If they are discarded, the data set decreases, making analysis difficult. Therefore we manipulated the missing values to prevent shrinkage of the data set and thus negatively affect the final result. We have filled in the missing data and will mention this in detail later. Table (1) shows the original data for the data set and Table (2) the updated data for the data set. The updated data includes 25,790 consumers, 2088 deviant users, and 23,702 impartial users. After processing the blank cells, we deleted all records with more than 10 blank cells and performed unbalanced data processing.

Table 2: Metadata information of the updated electricity theft dataset.

Description	Value
Time window of data collection	1 January 2014–31 October 2016
Total number of consumers	25790
Number of normal users	23702 approx. 91.19%
Number of aberrant users or electricity thieves	2088 approx. 8.81%
Missing data cases approx.	0.00000388%

Normal users have varied electricity usage habits. Figure (1) depicts a typical user and an electrical thief's monthly use. An aberrant or electricity-theft user has a fluctuating usage pattern.

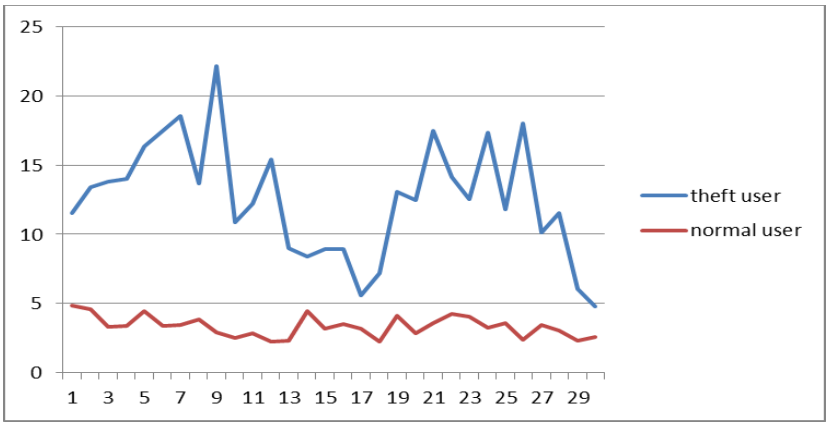

Fig. 1: monthly electric power consumption pattern.

The electricity theft dataset is an unbalanced dataset, where one class is much lower than the other. Figure (2) shows the distribution of users who are normal and thieves, Figure (3) shows their distribution in the updated dataset, and Figure (4) shows their distribution in a balanced updated dataset.

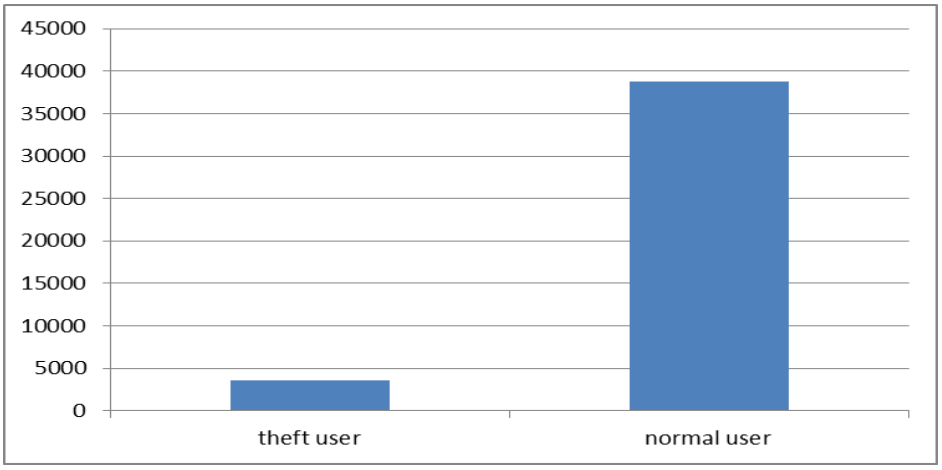

Fig. 2: Distribution of theft and normal users in the original dataset

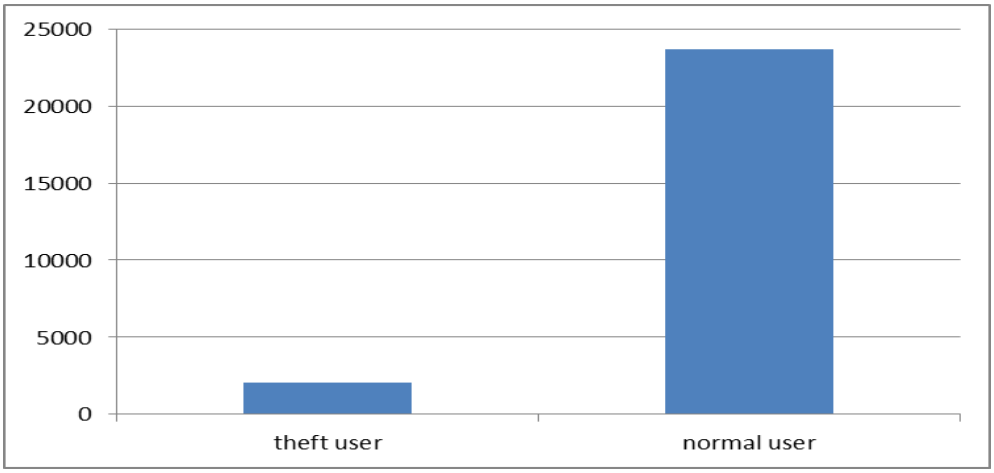

Fig. 3: Distribution of theft and normal users in the updated dataset

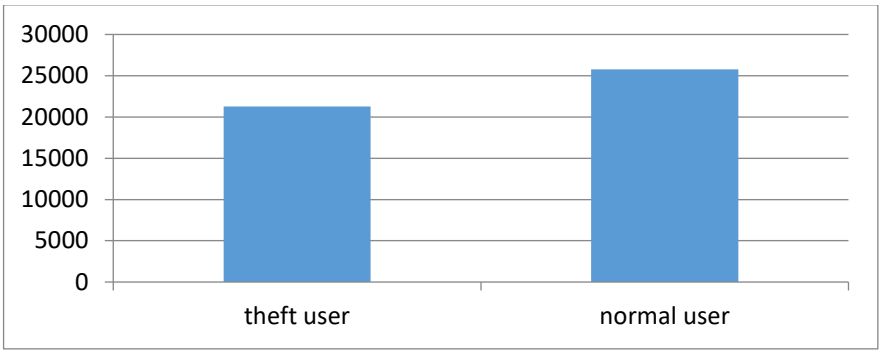

Fig. 4: Distribution of theft and normal users in a balanced updated dataset

3. The Proposed System Layout

This section explains the general layout of the proposed theft classification system. The proposed system consists of six main stages, as shown in Figure (5) and Figure (6) shows the description of the SGCC dataset.

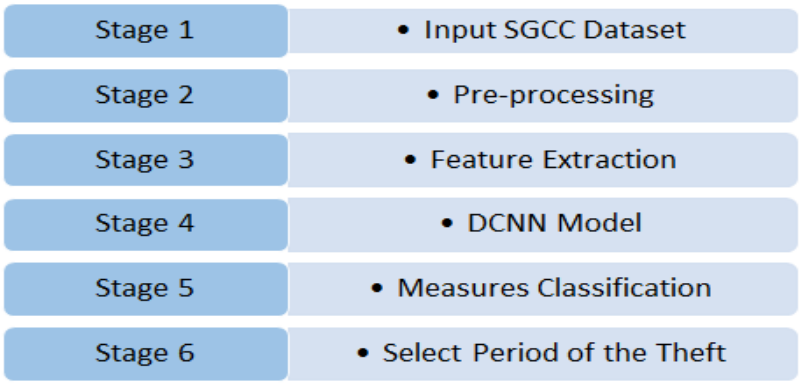

Fig. 5: The Proposed System Stages.

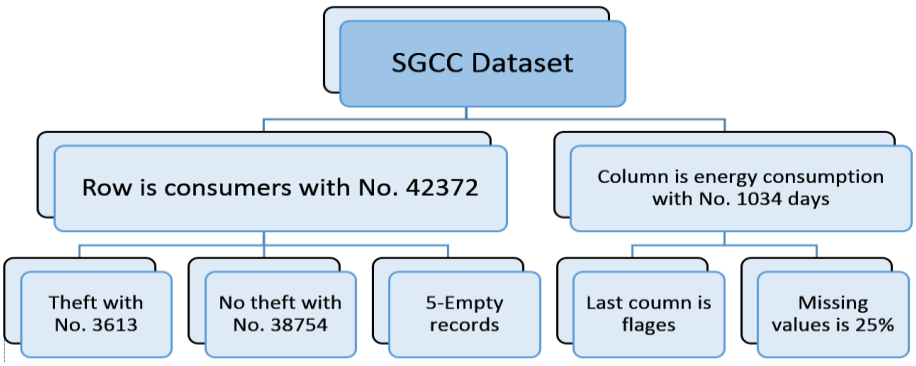

Fig. 6: SGCC Dataset description.

7.1 Pre-processing Stage

Primary data processing is one of the most important steps that affect the accuracy of the system, so it is necessary to filter the raw data and find the updated data. The feature extraction methods are applied to this updated data. Figure (7) shows the pre-processing steps of the SGCC dataset.

1. The columns of the dataset are ordered by date. This order is easy to deal with the dataset and determine the periods of the theft. The dataset history starts from January in the year 2014 to October in the year 2016.
2. Remove all empty records, where there are five empty records in the SGCC dataset. Empty records indexes are $(421, 686, 3030, 37832,$ and $41000)$. Three of these records are electricity theft (theft) and two of them records are natural electrical energy (no-theft).
3. Before the size empty reading, the centroid value is calculated for the closest neighbor. Find the M nearest non-empty readings at the before side (posb(1),posb(2),...,posb(M)) whose days numbers are Find the position and centroid value of the nearest non-empty neighbor:

$$C_b = \frac{1}{M}\sum_{i=1}^{M} R_b(i), \qquad P_b = \frac{1}{M}\sum_{i=1}^{M} Pos_b(i) \tag{1}$$

4. Following the size empty reading, the centroid value for the closest neighbor is calculated. Locate the M non-empty readings that are nearest to each other on the after side (Ra(1), Ra(2),..., Ra(M)) whose days numbers are (posa(1), posa(2),..., posa(M)). Find the position and centroid value of the nearest non-empty neighbor:

$$C_a = \frac{1}{M}\sum_{i=1}^{M} R_a(i), \qquad P_a = \frac{1}{M}\sum_{i=1}^{M} Pos_a(i) \quad (2)$$

5. (5) On the SGCC dataset, linear interpolation (LI) is used to process missing values. Using the centroids, the missing value R(i) is as follows:

$$R(i) = \frac{C_a - C_b}{P_a - P_b}(i - P_b) + C_b \quad (3)$$

6. Remove all records with more than ten cells missing values, where the missing data ratio is approx. is 0.00000388%. The total number of consumers is 25790. The number of normal users is 23702 approximation of 91.19%. The number of aberrant users or electricity thieves is 2088 approx. 8.81%.
7. We use ADASYN to acquire comprehensive SGCC data. In essence, SMOTE is improved upon. Minor change: it inserts new random values that are linearly connected with the parent samples but have a little bit larger variance after creating n-nearest neighbor samples. This tweak produces a more representative sampling of data. To produce more examples of the underrepresented classes, the ADASYN algorithm must first determine how many synthetic data samples, g, must be generated. The following equation may be used to determine this value:

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

where (mj, mi) is the number of representatives from the majority and minority groups, respectively. An equality constraint between the two groups, [0,1], is employed to determine the relative weight of each group. Then, using the Euclidean distance, we determine K's closest neighbors and use that information to compute the ratio ri as:

$$ri = \frac{\delta_i}{k}, \quad \text{where } i = 1 \ldots\ldots \quad (5)$$

To clarify, the number of samples from the majority class in the k nearest neighbors, and the number of synthetic samples; hence, r i is in the range [0, 1]. Last but not least, g i is the total of simulated data samples.

$$G_i = r_i * g \quad (6)$$

ADASYN serves a dual purpose: it avoids the model from being biased while also enhancing the classifier's learning performance for challenging theft scenarios.

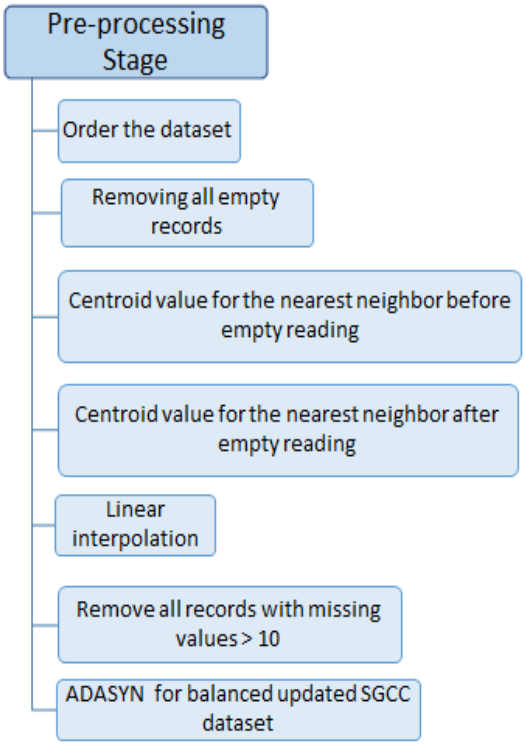

Fig. 7: SGCC pre-processing Steps.

3.2 Features Extraction Stage

The feature extraction is include many features are slope, average, moment, Stander Deviation (SD), total harmonic distortion, log energy, Peak to Peak (P2P), energy entropy, 5skewness, kurtosis, and Crest Factor (CF). All these features are explained in chapter two, where the first three measures are applied each month in the year for the SGCC dataset, and other measures are applied to all recorded for the SGCC dataset. Figure (8) shows the steps for extracting features through statistical measures of the SGCC dataset.

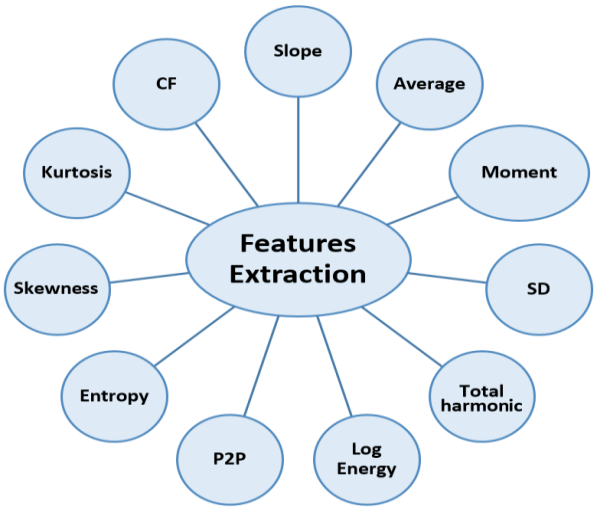

Fig. 8: The feature extraction by statistical measures.

3.3 Create DCNN Model Stage

The proposed CNN model was designed with several layers to classify the dataset. The input layer is a 2D image with a size of 13×19×1, whereas the original records are 1D with a size of 1×247. We convert a 1D record into a 2D image by using reshape function, whereas CNN is dealing with 2D-image. Figure (9) describes all Layers of the CNN Model.

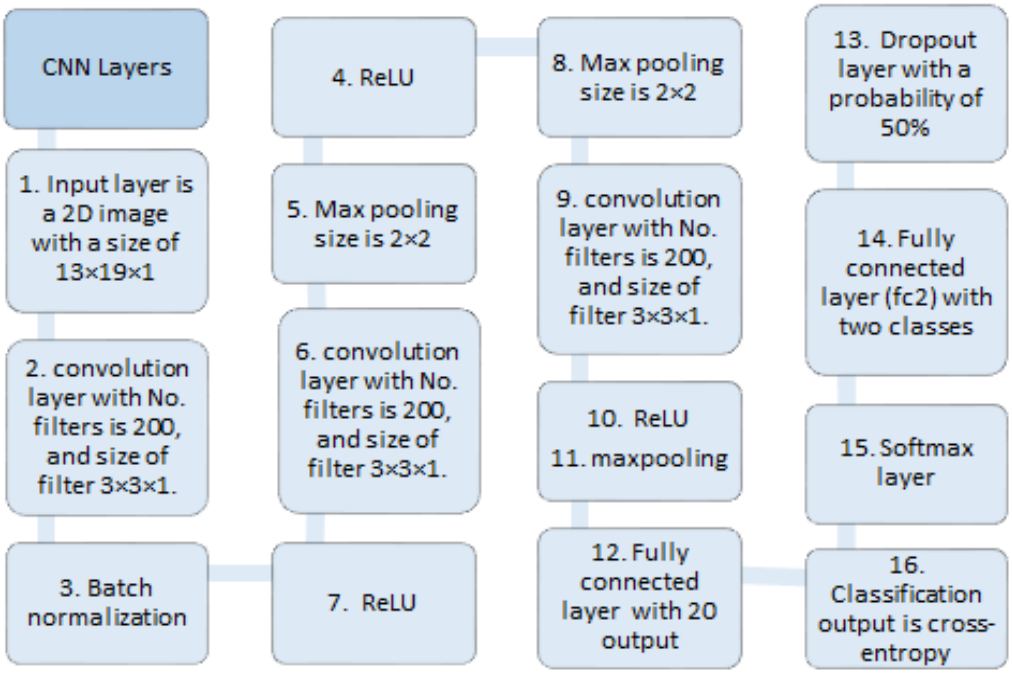

Fig. 9: Layers for CNN Model.

Tabel (3) shows the proposed CNN layers and parameters where the number of parameters is weights and bias, FS is filter size, the stride is a stride of filter, and NF is the number of filters, the number of filters computes as follows:

$$\text{No. of paramteres} = (f_{\text{width}} \times f_{\text{height}} \times \text{input}_{\text{channel}} + \text{bias}) \times N_{\text{filter}}$$

The output of the pooling layer compute as follows:

$$\text{size}_{\text{pooling}} = \frac{\text{image size} - \text{filter size}}{\text{stride}} + 1$$

Table 3: The proposed CNN layers and parameters.

Id	Layers	Input size	Output size	Filter Size(FS)	N. parameters
1.	Input layer	$19 \times 13 \times 1$	-	-	0
2.	Convolution layer 1	$19 \times 13 \times 1$	$19 \times 13 \times 200$	$FS = 3 \times 3$ NF=200	2000
3.	Batch normalization layer	$19 \times 13 \times 200$	$19 \times 13 \times 200$	-	0
4.	Activation Using ReLU (relu1)	$19 \times 13 \times 200$	$19 \times 13 \times 200$	-	0
5.	Max pooling layer (maxpool1)	$19 \times 13 \times 200$	$9 \times 6 \times 200$	$FS = 2 \times 2$ $Strid = 2 \times 2$	0
6.	Convolution layer (conv2)	$9 \times 6 \times 200$	$9 \times 6 \times 200$	3×3	360200
7.	ReLU (relu2)	$9 \times 6 \times 200$	$9 \times 6 \times 200$	-	0
8.	Max pooling layer (maxpool2)	$9 \times 6 \times 200$	$4 \times 3 \times 200$	$FS = 2 \times 2$ $Strid = 2 \times 2$	0
9.	Convolution layer (conv3)	$5 \times 4 \times 200$	$5 \times 4 \times 200$	3×3	360200
10.	ReLU (relu3)	$4 \times 3 \times 200$	$4 \times 3 \times 200$	-	0
11.	Max pooling layer (maxpool3)	$4 \times 3 \times 200$	$2 \times 1 \times 200$	-	0
12.	Fully connected layer (fc1)	$2 \times 1 \times 200$ =400	20	-	8000
13.	Dropout layer(0.5)	20	10	D=0.5	0
14.	Fully connected layer (fc2)	10	2	-	20
15.	Softmax layer	2	2	-	0
16.	Classification output	2	1	-	0

Experiments with the proposed CNN are conducted by first providing a sample data set, which is then subdivided into two distinct categories: designing and testing. The model parameters are initially defined using the training set, and the trained model is then tested using the validation data. After the training process is over, the model is put to the test on the test dataset. The goal of separating design data into training and validation is to set aside some of the data for later use in evaluating results. The training data partitioning is depicted in Figure (10).

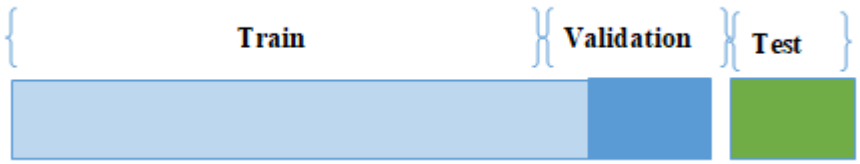

Fig. 10: The split of the data into training, validation, and test data.

After the training procedure is finished, the testing set is used to measure the efficiency of the final model. The test dataset is used to compute the different performance characteristics such as accuracy, sensitivity, specificity, and F-measure.

4. Compute the Energy Theft Period

This section explains and shows how to determine the period of power theft in the SGCC data set after processing the missing values and deleting the records with the largest 10 blank cells. We obtained a data set consisting of 2088 thieves and 23701 non-thieves records, Then we took the data of houses from the category of thieves, which numbered 2088, and applied algorithm (1) to it, and we got the total number of days and date in which the theft occurred for each house among the 2088 houses. Algorithm(1) explain the Compute the Energy Theft Period steps, Table (6) represents the total number of days in

which the theft occurred for five houses Only, the date is mentioned for the same days in Table (7), and finally, Figures (11, 12, 13, 14, 15) respectively... show the normal and abnormal consumption of electrical energy in five houses out of the 2088 houses over 1034 a day.

Algorithm (1): Date of energy theft

Input: daily consumption of SGCC dataset (A), where the first record is the date and the other record is daily consumption for each house, the number of records is N=2089, and the number of columns is M=1034.
Output: Date of the energy theft day (Date_Theft).
Begin:
Step 1: Create matrix (Date_Theft) with zeros and size (N-1), where
Step 2: The select different threshold is $\mathrm{th} = [0.75,\ 0.85]$, and the number of the threshold is $\mathrm{L} = 2$
Step 3: Normalization of records and find the date of day theft period as follows:

```
  For k=1: L
 For i =2 to N
      Norm_A(i,:) = A(i,:)/max(A(i,:))
    For j =1 to M
      If Norm_A(i,j) >= th then
            Date_Theft(i, j) = A(1,j)
        End if
         End for j
       End for i
      End for k
 End of algorithm
```

Table 4: The number of theft periods for five records

Homes	Numbers of period
Home1	13
Home2	6
Home3	3
Home4	1
Home5	1

Table 5: The dates of theft periods for five records

Homes	Numbers of period
Home1	"2016/6/25" "2016/7/8" "2016/7/17" "2016/7/23" "2016/7/26" "2016/7/27" "2016/7/28" "2016/7/29" "2016/7/30" "2016/8/1" "2016/8/5" "2016/8/8" "2016/8/25"
Home2	"2014/4/11" "2014/4/18" "2014/5/1" "2014/5/9" "2014/5/16" "2014/5/17"
Home3	"2016/2/6" "2016/2/7" "2016/2/8"
Home4	"2015/8/10"
Home5	"2015/8/10"

The below figure explains five homes representing five records, where the x-axis represented days and y axis represented energy. The maximum wave represented energy theft.

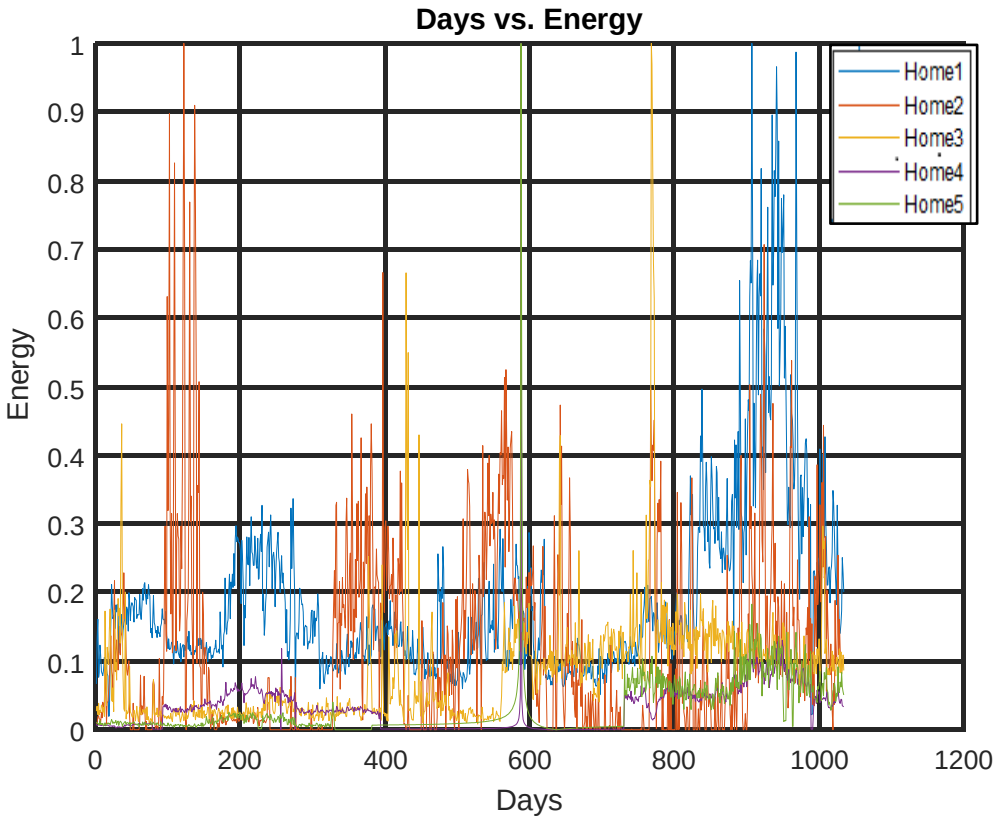

Fig. 11: Show the Energy consumption of Homes (1-5) in 1034 days.

The below figure explains one home represented one record, where the x-axis represented days and y axis represented energy. The maximum wave represented energy theft, where the number of theft days is 13 days when the threshold is 0.75.

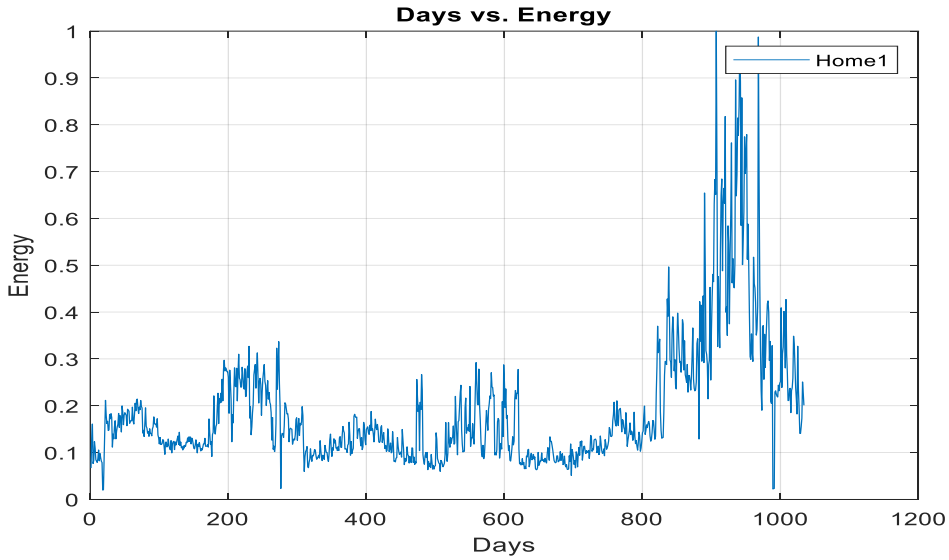

Fig. 12: show the Energy consumption of Home1 in 1034 days.

The below figure explains one home represented one record, where the x-axis represented days and y axis represented energy. The maximum wave represented energy theft, where the number of theft days is six days when the threshold is 0.75.

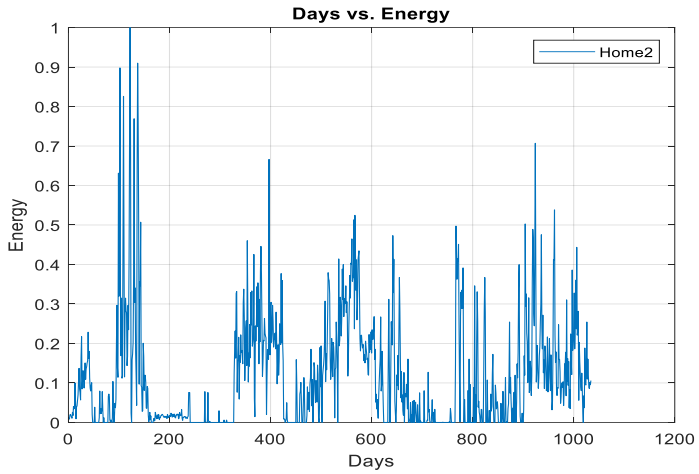

Fig. 13: show the Energy consumption of Home2 in 1034 days.

The below figure explains one home represented one record, where the x-axis represented days and y axis represented energy. The maximum wave represented energy theft, where the number of theft days is three days when the threshold is 0.75.

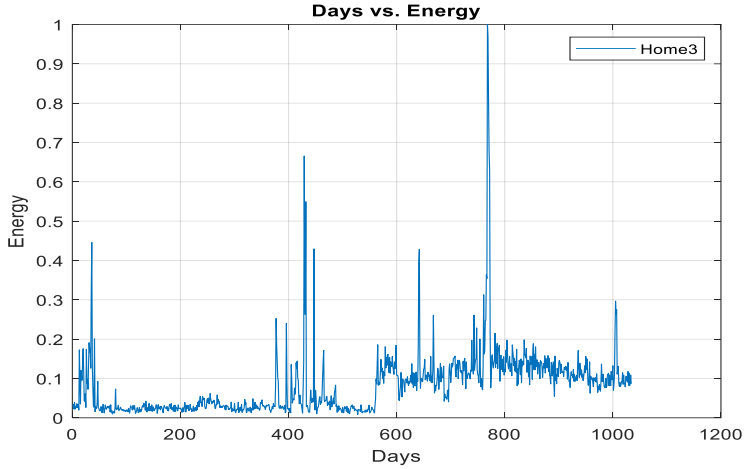

Fig. 14: show the Energy consumption of Home3 in 1034 days.

The below figure explains one home represented one record, where the x-axis represented days and y axis represented energy. The maximum wave represented energy theft, where the number of theft days is one day when the threshold is 0.75.

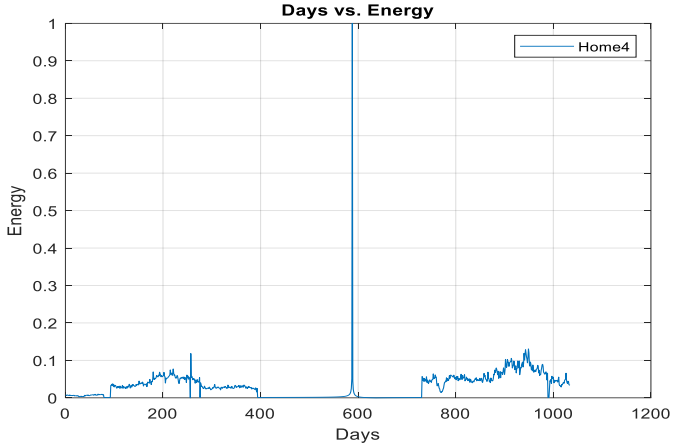

Fig. 15: show the Energy consumption of Home 4 in 1034 days**.**

The below figure explains one home represented one record, where the x-axis represented days and y axis represented energy. The maximum wave represented energy theft, where the number of theft days is one day when the threshold is 0.75.

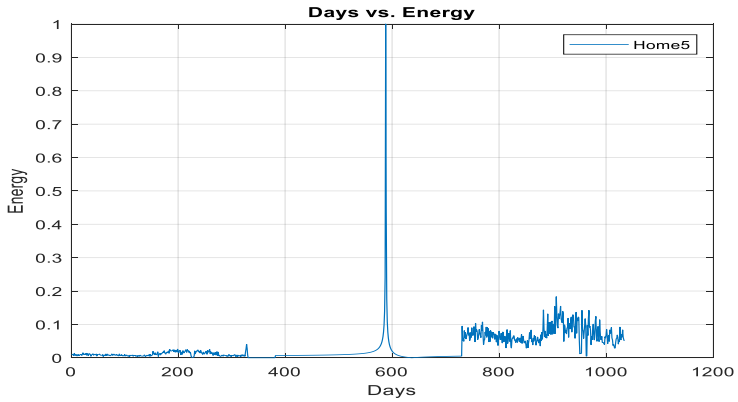

Fig. 16: Show the Energy consumption of Home5 in 1034 day.

5. Discuss the Results

Multiple cases have been studied to train different features and get different accuracy, as shown below:

Case 1: if some features were selected, such as average, standard division, and standard division/average, then accuracy was equal to 93.2.

Case 2: if some features are selected, such as average, slope, and moment, then accuracy equals 93.8.

Case 3: if some features are selected, such as average, Std Deviation, Variance, Peak to Peak, Energy Entropy, Skewness, Crest Factor, Total harmonic distortion, Log Energy, and Kurtosis; then attained accuracy was equal to 91.9.

Case 4: if using the feathers average, slope, and moment for every period and Std Deviation, Variance, Peak to Peak, Energy Entropy, Skewness, Crest Factor, Total harmonic distortion, Log Energy, and Kurtosis for all periods with gather, then attained accuracy was equal to 100.

Also, balance Dataset applied on same cases. Table (6) show the accuracy of the balance and imbalance dataset, Figure (16) shows the Confusion Matrix for the learning rate change, Figure **Error! No text of specified style in document.** shows the training process and loss rate learning rate change, and Table (7) shows the Measures of the training dataset and shows the result of the DCNN model and comparison with previous work in Table (8).

Table 6: Accuracy for balance and imbalance dataset.

Cases	Accuracy (Imbalance)	Accuracy (Balance)
1	93.2	88.4
2	93.8	88.8
3	91.9	63.2
4	100	91.8

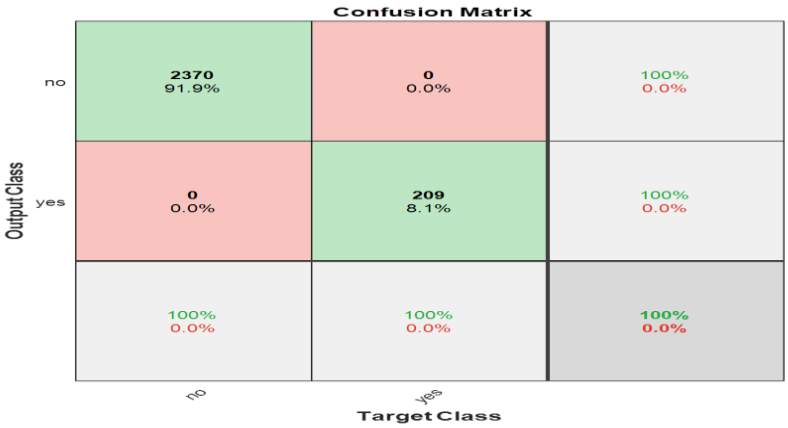

Fig. 16: Confusion Matrix for the learning rate change.

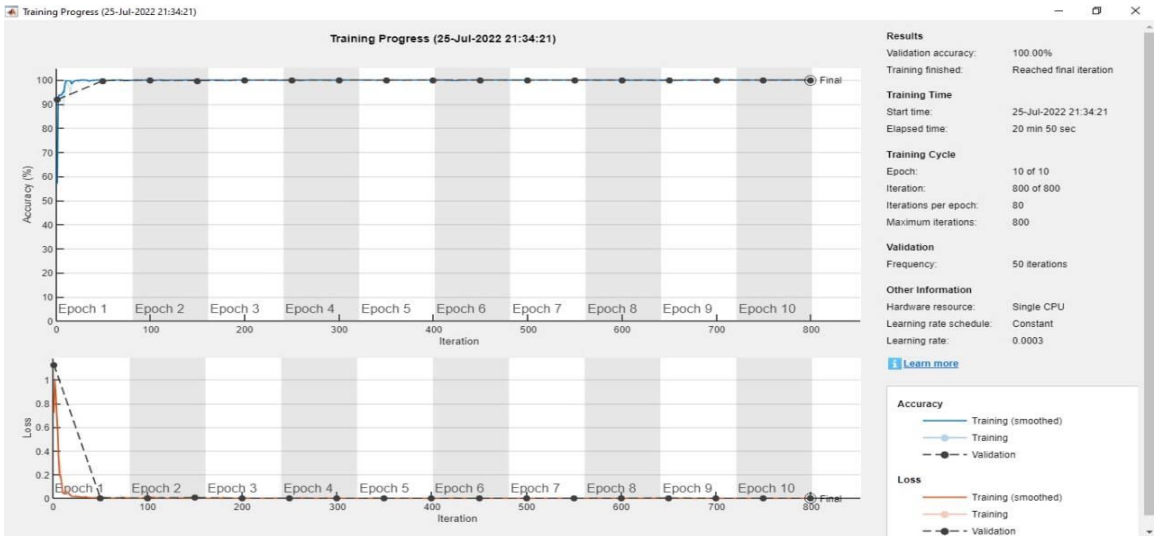

Fig. 17: Error! No text of specified style in document.**: training process and loss rate learning rate change.**

Table 7: Measures of the training dataset.

Parameters	Training method =ADAM; Mini Batch=256; Epochs=10; Learn Rate=3e-4
Accuracy	100
Precision	100
Recall	100
F-measure	100
False Negative Rate	0
False Positive Rate	0

6. The Comparison with the Previous Work

This section explains the comparison with other works as shown in table (5). Table (4) shows measures of the training dataset for optimal parameters. From the review of previous studies, we note the superiority of the proposed system by accuracy and other criteria such as recall, Precision, F-measure, False Negative Rate, and False Positive Rate.

Table 8: Summary of models, data sets, and performance measures.

Method	Year	Accuracy
Proposed DCNN model	2022	100
self-attention mechanism model [22]	2020	0.926
convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) [23]	2019	89
combined convolutional neural networks (CNNs) [24]	2019	92.67
hybrid DL model [25]	2022	88
CNN-GRU-PSO [26]	2020	87
CNN [27]	2019	93
Hybrid 2DCNN and BiLSTM [28]	2022	97
Ensemble Deep Convolutional Neural Network (EDCNN) algorithm.[29]	2020	99
autoencoder-bidirectional gated recurrent unit (AE-BiGRU) model [30]	2022	91.1
Autoregressive Integrated moving average (ARIMA). In the second stage, the distributed random forest (DRF) [31]	2022	98

7. Conclusion

The important conclusions of this paper are solving the problems of the SGCC dataset and finding feature extraction. SGCC dataset problems are missing values, empty records, Imbalanced data, and non-order. First, find feature extraction of the SGCC dataset by many methods, such as creating a new dataset for theft or non-theft classification. The features are

slope, average, moment, stander deviation, total harmonic distortion, Log Energy, Peak to Peak, Energy Entropy, Skewness, Kurtosis, and Crest Factor. Then applied DCNN to the classification dataset and computed accuracy. Finally, compare the proposed model with models for previous works. The number and date of the theft were then calculated for each of the records in which the theft occurred.

References

[1] R. R. Bhat, R. D. Trevizan, R. Sengupta, X. Li, and A. Bretas; "Identifying non-technical power loss via spatial and temporal deep learning", in Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), December 2016.

[2] Bohani F. A., Suliman, A., Saripuddin, M., Sameon, S. S., Md Salleh, N. S., & Nazeri, S.; "A comprehensive analysis of supervised learning techniques for electricity theft detection", *Journal of Electrical and Computer Engineering*, *2021*. (2021).

[3] Ibrahim Noor Sufyan Al-Janabi, and Belal Al-Khateeb. "Electricity-Theft Detection in Smart Grid Based on Deep Learning." Bulletin of Electrical Engineering and Informatics 10.4 (2021): 2285-2292.

[4] 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 of Things Journal, 6(5), 7659-7669.

[5] Hasan, M. N., Toma, R. N., Nahid, A. A., Islam, M. M., & Kim, J. M. (2019).lectricity theft detection in smart grid systems: A CNN-LSTM based approach. Energies, 12(17), 3310.

[6] Cheng, G., Zhang, Z., Li, Q., Li, Y., & Jin, W. (2021). Energy theft detection in an edge data center using deep learning. Mathematical Problems in Engineering, 2021.

[7] Abdulaal, M. J., Ibrahem, M. I., Mahmoud, M. M., Khalid, J., Aljohani, A. J., Milyani, A. H., & Abusorrah, A. M.," Real-Time Detection of False Readings in Smart Grid AMI Using Deep and Ensemble Learning"., IEEE Access, 10, 47541-47556., 2022.

[8] Khan, I. U., Javaid, N., Taylor, C. J., & Ma, X. (2022). Data Driven Analysis for Electricity Theft Attack-Resilient Power Grid. IEEE Transactions on Power Systems.

[9] Ullah, A., Javaid, N., Asif, M., Javed, M. U., & Yahaya, A. S. (2022). AlexNet, AdaBoost and Artificial Bee Colony Based Hybrid Model for Electricity Theft Detection in Smart Grids. IEEE Access, 10, 18681-18694.

[10] Alameady, M. H., Albermany, S., & George, L. E. (2022). Energy Theft Detection and Preventive Measures for IoT Using Machine Learning. Mathematical Statistician and Engineering Applications, 155-168.

[11] Zheng, Z., Yang, Y., Niu, X., Dai, H. N., & Zhou, Y. (2017). Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. IEEE Transactions on Industrial Informatics, 14(4), 1606-1615.

[12] Shuan Li , Yinghua Han , Xu Yao, Song Yingchen, Jinkuan Wang, and Qiang Zhao, Electricity Theft Detection in Power Grids with Deep Learning and Random Forests, Journal of Electrical and Computer Engineering Volume 2019, Article ID 4136874, 12 pages.

[13] Hussain F., Hussain R. Hassan, S. A. & Hossain E.," Machine learning in IoT security: Current solutions and future challenges". IEEE Communications Surveys & Tutorials, 22(3), 1686-1721, 2020.

[14] Hu W., Yang, Y. Wang, J. Huang, X. & Cheng Z.," Understanding electricity-theft behavior via multi-source data". In Proceedings of The Web Conference 2020 (pp. 2264-2274), 2020, April.

[15] Finardi P., Campiotti I., Plensack G., de Souza R. D., Nogueira R., Pinheiro, G. & Lotufo R.," Electricity theft detection with self-attention",. arXiv preprint arXiv:2002.06219, 2020.

[16] Ullah, A.; Javaid, N.; Samuel, O.; Imran, M.; Shoaib, M. CNN and GRU based deep neural network for electricity theft detection to secure smart grid. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020; pp. 1598–1602.

[17] Asif, M., Nazeer, O., Javaid, N., Alkhammash, E. H., & Hadjouni, M. (2022). Data Augmentation Using BiWGAN, Feature Extraction and Classification by Hybrid 2DCNN and BiLSTM to Detect Non-Technical Losses in Smart Grids. IEEE Access, 10, 27467-27483.

[18] Rouzbahani, H. M., Karimipour, H., & Lei, L. (2020, October). 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.

[19] Javaid, N., Qasim, U., Yahaya, A. S., Alkhammash, E. H., & Hadjouni, M. (2022). Non-technical losses detection using autoencoder and bidirectional gated recurrent unit to secure smart grids. IEEE Access.

[20] Badawi, S. A., Guessoum, D., Elbadawi, I., & Albadawi, A. (2022). A Novel Time-Series Transformation and Machine-Learning-Based Method for NTL Fraud Detection in Utility Companies. Mathematics, 10(11), 1878.