ELECTRICITY THEFT DETECTION IN SMART GRIDS BASED ON ARTIFICIAL NEURAL NETWORK.

Dhil Rohith B[1], Meganamani T[2], Deepak S[3], Chandru K S[4]

Student(1), Student(2), Student(3), Faculty(4)

Dhilrohith.cb20@bitsathy.ac.in [1], meganamani.cb20@bitsathy.ac.in [2], Deepak.cb20@bitsathy.ac.in [3], chandruks@bitsathy.ac.in [4]

Bannari Amman Institute of Technology, Sathyamangalam.

Abstract:

Smart grids are gaining popularity due to their ability to enhance security, reduce power outages, and improve energy efficiency, aligning with the increasing demand for electricity. However, a major challenge faced by smart grid operators is electricity theft, which leads to significant financial losses for utility companies. Therefore, electric power distribution firms are deeply concerned about this issue. This study aims to propose an effective method, leveraging artificial neural networks (ANNs), for detecting power theft in smart grids. The method involves utilizing a dataset on electricity usage sourced from Kaggle, which will undergo preprocessing before being fed into the ANN. The ANN will be trained on a dataset representing normal consumption patterns and tested on data containing instances of power theft. Performance evaluation will be based on metrics such as recall, accuracy, precision, and F1-score. The proposed method achieved promising results with 99% training and validation accuracies. To enhance usability, a Flask Web framework was employed to develop a user-friendly interface for outcome prediction. This research aims to provide a valuable tool for detecting electricity theft in smart grids, thereby improving security and revenue collection for utility companies. Furthermore, the techniques developed in this project can be extended to other domains requiring anomaly detection in large datasets, such as intrusion detection in computer networks and fraud detection in financial systems.

Keywords: Artificial Neural Network (ANN), Electricity Theft, Machine Learning, Minimum Redundancy maximum Relevance, Smart Grids.

Introduction:

The evolution of smart grid technology has revolutionized the landscape of energy distribution and management, offering a myriad of advantages such as heightened security, minimized power interruptions, and enhanced energy efficiency. Despite these considerable advancements, the persistent issue of electricity theft continues to plague utility companies, resulting in substantial revenue losses and undermining the integrity of the system itself. Consequently, electric power distribution firms are increasingly prioritizing the mitigation of electricity theft, recognizing it as a critical challenge that demands urgent attention.

This project aims to address the pressing problem of electricity theft in smart grids by leveraging Artificial Neural Networks (ANN). Our objective is to develop a robust and proactive strategy capable of effectively detecting and flagging instances of electricity theft. By harnessing the power of ANN technology, we seek to create a

sophisticated approach that enables utility companies to protect their revenue streams and bolster the security of smart grid infrastructure, thereby ensuring the continued reliability and sustainability of the energy distribution system.

The current system utilizes a DNN-based method for efficiently detecting electricity theft by analyzing carefully selected information. It demonstrates that incorporating frequency-domain features enhances classification accuracy compared to solely relying on time-domain features. Using a realistic dataset provided by the State Grid Corporation of China (SGCC), the system employs Principal Component Analysis (PCA) to reduce the feature space for better comprehension of results and future training optimization. To discern the significance of features and the superiority of frequency-domain over time-domain features in detecting theft, the system employs the Minimum Redundancy Maximum Relevance (mRMR) technique.

We present a systematic approach aimed at detecting electricity theft within smart grids, utilizing electricity usage datasets sourced from reputable platforms like Kaggle. To ensure the suitability of these datasets for analysis, we employ thorough preprocessing techniques. The refined data is then fed into an artificial neural network (ANN) framework, where the network undergoes training to identify intricate consumption patterns and irregularities.

The foundation of our methodology lies in building a robust ANN model, which addresses existing limitations through a comprehensive three-phase process: Preprocessing and Data Analysis, Classification, and Feature Extraction. We begin by meticulously preprocessing the electricity usage datasets, encompassing techniques such as feature extraction, normalization, and data cleaning. To overcome challenges in obtaining labeled data, we utilize Agglomerative clustering to categorize usage cases as loyal or unfaithful.

Our methodology employs cluster analysis, particularly Agglomerative clustering, to detect atypical usage patterns indicative of energy theft. This step is pivotal in generating labeled data essential for training the ANN model to discern complex theft patterns in electricity consumption data. Performance evaluation of our approach involves a diverse range of metrics including accuracy, recall, F1-score, and precision, to gauge its effectiveness comprehensively.

Moreover, we enhance accessibility and usability by developing a user-friendly web interface using the Flask Web framework. This interface facilitates easier interaction with the electricity theft detection system, empowering utility firms with a practical tool for bolstering grid security.

The primary aim of our project is to propose a novel methodology based on Artificial Neural Networks for electricity theft detection within smart grids. Leveraging state-of-the-art data analysis and machine learning techniques, our proposed system endeavors to significantly enhance the efficacy and reliability of electricity theft detection by addressing the limitations of current methods. Utility firms stand to benefit from the successful implementation of this methodology, strengthening revenue collection operations and fortifying the security of smart grid assets.

Advantage of Proposed System:

High Accuracy: Research has shown that artificial neural network (ANN) models achieve a high level of accuracy in detecting electricity theft. This is attributed to their ability to uncover intricate connections and patterns within consumption data, which may be difficult to detect using traditional statistical methods.

Robustness: ANN models are capable of handling noisy and incomplete data commonly encountered in real-world smart grid deployments. This inherent robustness makes ANN models less susceptible to errors and false positives.

Adaptability: ANN models possess the capability to adapt to changes within the smart grid environment, such as the emergence of new theft methods or shifts in usage patterns. This adaptability makes ANN models well-suited for the dynamic nature of smart grids.

Speed: ANN models excel in processing large volumes of data swiftly, enabling real-time detection of electricity theft. This rapid processing capability enables utility companies to promptly respond and implement corrective measures to mitigate revenue losses.

Automation: ANN models can be trained to automatically identify instances of electricity theft, eliminating the need for manual inspection and reducing the workload for utility companies. This automation results in significant cost savings and enhanced operational efficiency.

Methodology:

System Architecture:

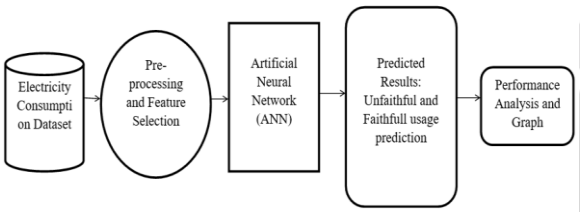


Fig.1

Data Collection:

In the first module, we undertake the task of data collection, which marks the initial phase in constructing a machine learning model. This stage holds significant importance as it directly impacts the model's performance; acquiring ample and high-quality data leads to greater accuracy in the model's predictions. Various methods exist for data gathering, such as manual interventions and web scraping, among others. The dataset used in this project is stored within the model folder and was obtained from the reputable online platform Kaggle.

Dataset:

The dataset comprises 3,510,433 individual entries, with nine columns providing the following information:

LCLid: Identification number associated with each entry.

day: Date in the format dd/mm/yyyy.

energy_mean: Average energy value.

energy_median: Median energy value.

energy_std: Standard deviation of energy values.

energy_max: Maximum energy value.

energy_sum: Total energy value.

energy_count: Count of energy entries.

energy_min: Minimum energy value.

Importing the necessary libraries:

We'll utilize Python as our primary language for this task. To start, we'll import necessary libraries, including pandas, numpy, matplotlib, and tensorflow. Additionally, we'll incorporate keras for building the primary model, sklearn for splitting the data for training and testing, and PIL for converting images into arrays of integers.

Clustering (To find Electricity Theft (Target value)) :

In our prior analysis, we employed Agglomerative clustering with a cluster score of 3 to assess the extent of electricity theft, focusing on mean energy values.

Splitting the dataset:

Divide the dataset into training and testing sets, with 20% allocated for testing and 80% for training.

Neural network:

In the second stage, a neural network is chosen to serve as the representation of the classification function. It includes:

A scaling layer tailored for classification tasks.

A layer comprised of perceptrons.

A layer dedicated to probability estimation.

The scaling layer is configured to incorporate both minimum and maximum scaling mechanisms. For the initial setup, a logistic activation function is employed, along with a single perceptron layer consisting of three neurons. Neural networks belong to a category of algorithms designed to recognize patterns within datasets, mimicking the functionality of the human brain. They find applications in various domains including regression, classification, and image recognition, among others.

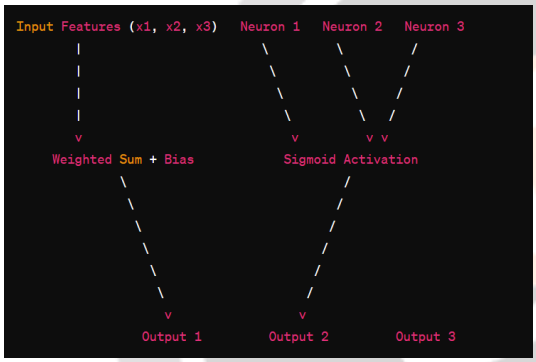


Fig.2

Artificial neural networks aim to mimic the structure and functions of the human brain, but they exhibit both similarities and disparities. One notable distinction is in how data is processed: artificial neural networks handle data sequentially, whereas biological neural networks process data concurrently. Moreover, artificial neural networks boast a faster processing speed, typically measured in nanoseconds, compared to the slower processing speed of biological neural networks, which are typically measured in milliseconds.

Architecture of ANN:

A neural network earns the title of a multi-layer perceptron due to its composition of multiple layers, each serving distinct functions. As the complexity of the model escalates, so does the number of layers within the network.In its fundamental form, a neural network comprises three layers: an input layer, a hidden layer, and an output layer. Before addressing a specific problem, these networks must undergo training using training data and machine learning techniques. The input layer receives input signals, transmits them to subsequent layers, and ultimately generates the final prediction.

Now, let's delve into the concept of perceptron.

Perceptrons, or dense layers, are integral components of multi-layer perceptrons as previously elucidated. They predominantly consist of neurons, which are fundamental units that amalgamate to form perceptrons. Essentially, perceptrons, or dense layers, are depicted as a vertical arrangement of neurons, as illustrated in the accompanying image, where each circle symbolizes a neuron. Each neuron in this illustration possesses a weight (e.g., w1, w2, w3) and a bias. Computation entails utilizing the formula combination = bias + weights * input (e.g., F = w1x1 + w2x2 + w3*x3), followed by the application of an activation function, resulting in output =

activation(combination). In the depicted example, the sigmoid activation function is represented as 1/(1 + e^-F). Other common activation functions include ReLU, Leaky ReLU, and tanh, among others.

Working of ANN

Initially, information is transmitted to the input layer, which subsequently transfers it to the hidden layers. Within these layers, each input is initially endowed with a random weight through the interconnections. Following this, bias is introduced into every input neuron. The weighted sum, encompassing both weights and bias, is then subjected to processing by the activation function. This function determines the activation of nodes, facilitating feature extraction and eventual output computation. This entire process is termed as Forward Propagation.

In Backward Propagation, weights are adjusted to minimize error after obtaining the output model and comparing it with the original output to assess error. This iterative process is repeated for a predetermined number of epochs or iterations. Ultimately, upon completion of predictions, model weights undergo modification.

Model selection:

In this module, an Artificial Neural Network (ANN) model is created and trained using the extracted features and labels, distinguishing between faithful and unfaithful cases. The complexity of the neural network, which optimizes generalization performance, is determined through order selection. This entails finding the number of neurons that minimize inaccuracies in the selection cases.

Additionally, input selection, also known as feature selection, is performed to identify the set of inputs that yield the best generalization. Alt
hough a genetic algorithm is utilized for this purpose, it fails to decrease the selection error value, prompting the retention of all input variables.
The Keras model is defined by a series of layers added sequentially. The input_dims argument specifies the input when creating the first layer, which in this case consists of 8 input features.

Determining the appropriate number of layers and neurons in each layer presents a challenge. The Keras tuner, which explores various layer configurations, neuron counts, and activation functions, is commonly used for this task. However, this approach can be time-consuming due to the exhaustive combination and permutation search. For more detailed information, refer to the Keras Tuner documentation.

In this example, the Dense class is utilized to define a fully connected, three-layer network. The number of neurons in each layer is specified in the first argument, while the activation function is specified in the activation argument. For this binary classification problem, ReLU is employed as the activation function in the first two layers, while sigmoid is used in the final layer.

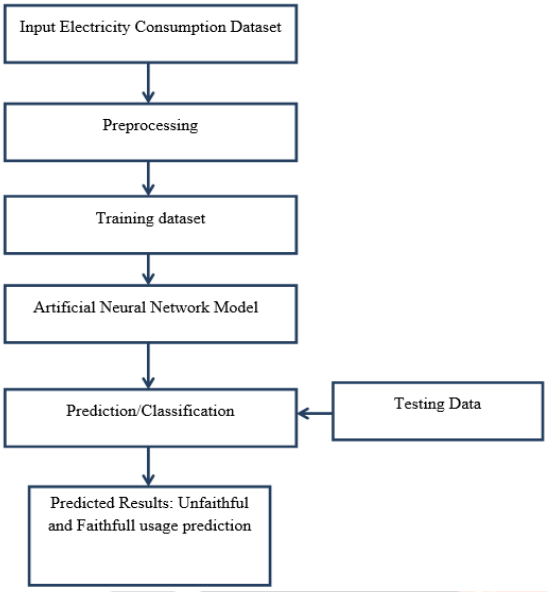
During compilation, the optimizer for updating weights, as well as any metrics and the loss function for calculating errors, must be specified. For binary classification tasks like this one, "binary_crossentropy" is typically used as the loss argument. The optimizer "Adam" is chosen for its ability to automatically adjust and yield favorable results across various scenarios. Classification accuracy is evaluated and reported using the metrics parameter.

Next, the model is fitted to the loaded data using the fit() function, utilizing the dataset for a specified number of iterations, or epochs. The batch_size parameter determines the number of dataset rows updated within each epoch. Here, a batch size of 64 and 100 epochs are chosen.

To evaluate the model's performance, the evaluate() function assesses it on the dataset, requiring input and output arguments. It generates predictions for each input-output pair and collects scores, including average loss and relevant metrics like accuracy. The first item in the returned list from evaluate() represents the model's loss, while the second indicates its accuracy on the dataset. The loss value is disregarded here, as the focus is solely on reporting accuracy.

Finally, predictions are made using the predict() function on the model. Since the output layer employs a sigmoid activation function, the predictions range between 0 and 1, representing probabilities.

Flowchart



:

Apply the model and plot the graphs for accuracy and loss:

We'll utilize the fit function to compile and apply the model, with a batch size of 64. Subsequently, we'll plot graphs depicting accuracy and loss. On average, we attained a training accuracy of 99%.

Analyze and Prediction:

In this module, relevant features are extracted from the processed data to serve as inputs for the ANN. Specifically, we select 7 features from the dataset:
energy_median: Median energy value
energy_mean: Mean energy value
energy_max: Maximum energy value
energy_count: Count of energy entries
energy_std: Standard deviation of energy values
energy_sum: Total energy value
energy_min: Minimum energy value
target:"Faithful" and "Unfaithful"

Accuracy on test set:

In this module, we will evaluate the accuracy and effectiveness of the developed ANN model using a test dataset. We attained a test set accuracy of 99%.

Saving the Trained Model:

Initially, ensure that the trained and tested model is saved into either a .h5 or .pkl file using a library such as pickle, once you're confident in deploying it to a production-ready environment. Confirm that pickle is installed within your setup. Then, proceed to import the module and export the model to a .pkl file. Finally, utilize the Flask web framework to implement the trained ANN model in a real-world scenario for electricity theft detecti

Fig.3

Conclusion:

This paper explores the utilization of artificial neural networks (ANNs) for detecting electricity theft within smart grids. The study reveals that classification using ANNs outperforms the current system, achieving a remarkable 99% Training Accuracy and 99% Validation Accuracy. The proposed approach leverages consumption data trends, offering potential applications in anomaly detection across various domains beyond power distribution networks. While our work primarily focuses on gradual theft detection, it contributes to improving energy theft detection accuracy. In summary, the proposed ANN-based approach holds promise in significantly mitigating revenue losses due to energy theft in smart grids. By identifying theft incidents in real-time and notifying utility providers, the solution has the potential to diminish the impacts of electricity theft while enhancing overall grid security and efficiency.

Future Work:

For future endeavors, we aim to enhance our technique by incorporating real-time electricity theft detection capabilities. Further validation against datasets from diverse sources will ensure the method's applicability beyond the specific consumption patterns observed in SGCC consumer data.

References:

[1] S. Foster. (Nov. 2, 2021). Non-Technical Losses: A $96 Billion Global Opportunity for Electrical Utilities. [Online]. Available: https://energycentral.com/c/pip/ non-technical-losses-96-billion-globalopportunity-electrical-utilities

[2] Q. Louw and P. Bokoro, ``An alternative technique for the detection and mitigation of electricity theft in South Africa," SAIEE Afr. Res. J., vol. 110, no. 4, pp. 209216, Dec. 2019.

[3] M. Anwar, N. Javaid, A. Khalid, M. Imran, and M. Shoaib, ``Electricity theft detection using pipeline in machine learning," in Proc. Int. Wireless Commun. Mobile Comput. (IWCMC), Jun. 2020, pp. 21382142.

[4] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou, ``Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," IEEE Trans. Ind. Informat., vol. 14, no. 4, pp. 16061615, Apr. 2018.

[5] P. Pickering. (Nov. 1, 2021). E-Meters Offer Multiple Ways to Combat Electricity Theft and Tampering. [Online]. Available: https://www.electronicdesign.com/technologies/meters

[6] X. Fang, S. Misra, G. Xue, and D. Yang, ``Smart gridThe new and improved power grid: A survey," IEEE Commun. Surveys Tuts., vol. 14, no. 4, pp. 944980, 4th Quart., 2012.

[7] M. Ismail, M. Shahin, M. F. Shaaban, E. Serpedin, and K. Qaraqe, ``Efficient detection of electricity theft cyber attacks in AMI networks," in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), Apr. 2018, pp. 16.

[8] A. Maamar and K. Benahmed, ``Machine learning techniques for energy theft detection in AMI," in Proc. Int. Conf. Softw. Eng. Inf. Manage. (ICSIM), 2018, pp. 5762.

[9] A. Jindal, A. Schaeffer-Filho, A. K. Marnerides, P. Smith, A. Mauthe, and L. Granville, ``Tackling energy theft in smart grids through data-driven analysis," in Proc. Int. Conf. Comput., Netw. Commun. (ICNC), Feb. 2020, pp. 410414.

[10] I. Diahovchenko, M. Kolcun, Z. □onka, V. Savkiv, and R. Mykhailyshyn, ``Progress and challenges in smart grids: Distributed generation, smart metering, energy storage and smart loads," Iranian J. Sci. Technol., Trans. Electr. Eng., vol. 44, no. 4, pp. 13191333, Dec. 2020.

[11] M. Jaganmohan. (Mar. 3, 2022). Global Smart Grid Market Size by Region 20172023. [Online]. Available: https://www.statista.com/statistics/246154/global-smart-grid-marketsize-by-region/

[12] Z. Zheng, Y. Yang, X. Niu, H.-N. Dai, and Y. Zhou. (Sep. 30, 2021). Electricity Theft Detection, [Online]. Available: https://github.com/henryRDlab/ ElectricityTheftDetection

[13] D. O. Dike, U. A. Obiora, E. C. Nwokorie, and B. C. Dike, ``Minimizing household electricity theft in Nigeria using GSM based prepaid meter," Amer. J. Eng. Res., vol. 4, no. 1, pp. 5969, 2015.
[14] P. Dhokane, M. Sanap, P. Anpat, J. Ghuge, and P. Talole, ``Power theft detection & initimate energy meter information through SMS with auto power cut off," Int. J. Current Res. Embedded Syst. VLSI Technol., vol. 2, no. 1, pp. 18, 2017.
[15] S. B. Yousaf, M. Jamil, M. Z. U. Rehman, A. Hassan, and S. O. G. Syed, ``Prototype development to detect electric theft using PIC18F452 microcontroller," Indian J. Sci. Technol., vol. 9, no. 46, pp. 15, Dec. 2016.
[16] K. Dineshkumar, P. Ramanathan, and S. Ramasamy, ``Development of ARM processor based electricity theft control system using GSM network," in Proc. Int. Conf. Circuits, Power Comput. Technol. (ICCPCT), Mar. 2015, pp. 16.
[17] S. Ngamchuen and C. Pirak, ``Smart anti-tampering algorithm design for single phase smart meter applied to AMI systems," in Proc. 10th Int. Conf. Electr. Eng./Electron., Comput., Telecommun. Inf. Technol., May 2013, pp. 16.
[18] B. Khoo and Y. Cheng, ``Using RFID for anti-theft in a Chinese electrical supply company: A cost-benet analysis," in Proc. Wireless Telecommun. Symp. (WTS), Apr. 2011, pp. 16.
[19] J. Astronomo, M. D. Dayrit, C. Edjic, and E. R. T. Regidor, ``Development of electricity theft detector with GSM module and alarm system," in Proc. IEEE 12th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ., Manage. (HNICEM), Dec. 2020, pp. 15.
[20] P. Jokar, N. Arianpoo, and V. C. M. Leung, ``Electricity theft detection in AMI using customers' consumption patterns," IEEE Trans. Smart Grid, vol. 7, no. 1, pp. 216226, Jan. 2015.
[21] W. Han and Y. Xiao, ``A novel detector to detect colluded non-technical loss frauds in smart grid," Comput. Netw., vol. 117, pp. 1931, Apr. 2017.
[22] S. Sahoo, D. Nikovski, T. Muso, and K. Tsuru, ``Electricity theft detection using smart meter data," in Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT), Feb. 2015, pp. 15.
[23] S. K. Singh, R. Bose, and A. Joshi, ``PCA based electricity theft detection in advanced metering infrastructure," in Proc. 7th Int. Conf. Power Syst. (ICPS), Dec. 2017, pp. 441445.
[24] M. Di Martino, F. Decia, J. Molinelli, and A. Fernández, ``Improving electric fraud detection using class imbalance strategies," in Proc. ICPRAM, 2012, pp. 135141.
[25] M. N. Hasan, R. N. Toma, A.-A. Nahid, M. M. M. Islam, and J.-M. Kim, ``Electricity theft detection in smart grid systems: A CNN-LSTM based approach," Energies, vol. 12, no. 17, p. 3310, 2019.
[26] Y. Peng, Y. Yang, Y. Xu, Y. Xue, R. Song, J. Kang, and H. Zhao, ``Electricity theft detection in AMI based on clustering and local outlier factor," IEEE Access, vol. 9, pp. 107250107259, 2021.
[27] A. Aldegheishem, M. Anwar, N. Javaid, N. Alrajeh, M. Shaq, and H. Ahmed, ``Towards sustainable energy efciency with intelligent electricity theft detection in smart grids emphasising enhanced neural networks," IEEE Access, vol. 9, pp. 2503625061, 2021.
[28] K. Phil, MATLAB Deep Learning: With Machine Learning, Neural Net-works and Articial Intelligence. Seoul, South Korea: Apress, 2017.
[29] S. Notley and M. Magdon-Ismail, ``Examining the use of neural networks for feature extraction: A comparative analysis using deep learning, support vector machines, and K-nearest neighbor classiers," 2018, arXiv:1805.02294.
[30] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, ``Artificial neural networks-based machine learning for wireless networks: A tutorial," 2017, arXiv:1710.02913.
[31] D. Jurafsky and J. H. Martin, Speech and Language Processing: An Introduction to Speech Recognition, Computational Linguistics and Natural Language Processing, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2020.
[32] L. Deng, ``A tutorial survey of architectures, algorithms, and applications for deep learning," APSIPA Trans. Signal Inf. Process., vol. 3, no. 1, pp. 129, 2014.
[33] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, ``Efficient processing of deep neural networks: A tutorial and survey," Proc. IEEE, vol. 105, no. 12, pp. 22952329, Dec. 2017.