International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Utilizing Machine Learning for Detecting Electricity Theft in Smart Grids

M.Yasmi[1], Mr.K. Praveen Kumar[2], Dr.D. Jaya Kumari[3]

[1]M. Tech Student, Department of Computer Science, Sri Vasavi Engineering College (A), Pedatadepalli, Tadepalligudem– 534101.
[2]Assistant Professor, Department of CSE, Sri Vasavi Engineering College(A), Pedatadepalli, Tadepalligudem– 534101.
[3]Professor & HOD, Department of CSE, Sri Vasavi Engineering College(A), Pedatadepalli, Tadepalligudem– 534101.

Abstract:
Smart grids, which provide several advantages, such as improved energy efficiency, less power outages, and higher security, are growing in popularity as a result of the rising need for electricity. But one of the biggest problems with smart grids is power theft, which costs utility companies a lot of money. Therefore, electric power distribution firms are quite concerned about theft of power. The purpose of this study is to offer an effective technique based on artificial neural networks (ANNs) for identifying Smart grid electricity theft. After training on a dataset of acceptable consumption patterns, the ANN model will be evaluated on information about energy theft events. The design will be tested using test data in order to assess the effectiveness of the suggested strategy. The outcomes that we anticipated from our suggested ANN-based method for Smart grid detection of electricity theft are favourable. 99% Training Accuracy and 99% Validation Accuracy were attained by our method. The performance measures that will be employed include F1-score, recall, accuracy, and precision. Additionally, we created the proposed system that makes use of the Flask Web framework to make it easier to use and provide a better user interface for outcome prediction. The research will likely produce an efficient method for employing ANN to identify energy theft in smart grids, which utility companies can utilise to increase revenue collection and fortify the security of the smart grid. This research may potentially be expanded to other fields, such intrusion detection in computer networks and fraud detection in financial systems, that entail anomaly identification in large-scale datasets.

Keywords: Artificial Neural Network, Flask Web Framework, Smart Grids, Energy Theft, large-scale.

1. Introduction

Artificial neural networks (ANNs) replicate the mind's complex neuronal association. Arrows show output connections among synthetic neurons, which can be nodes. ANNs, or connection systems, are animal brain neural community-based computational models. rather than developing undertaking-specific policies, those structures "examine" from examples. A neural network (ANN) is made of "artificial neurons," which might be modeled after real brain neurons. artificial neurons ship alerts like mind synapses. artificial neurons method alerts and ship them to other neurons. maximum ANN implementations deal with indicators as the output of each artificial neuron is a non-linear characteristic

of it's enter sum. synthetic neurons' "edges," or connections, change weight all through learning, affecting sign strength. artificial neurons may also have layer-primarily based thresholds that decide signal transmission. signals can also pass thru multiple layers from the layer of input to the layer of output, where it will be converted. An ANN operates as a gadget of interconnected nodes, closely corresponding to the intricate arrangement of neurons within the human brain. It permit's delve into the key additives and functioning of ANNs:

Nodes (Artificial Neurons):

- Each node in an ANN represents an artificial neuron. These nodes process and transmit information.
- Connections between nodes are depicted by arrows, signifying the flow of output between neurons.

Inspiration from Biological Neural Networks:

- ANNs, also known as connectionist systems, draw inspiration from animal brain neural networks.
- Unlike traditional rule-based programming, ANNs "learn" tasks by analyzing examples.

Structure of an ANN:

- Artificial neurons function similarly to synapses in the brain, transmitting information or signals.
- An artificial neuron processes impulses and transmits them to associated neurons.

Signal Representation and Processing:

- Signals are represented as real numbers in most ANN implementations.
- A non-linear function of the total input is the output of each artificial neurone.
- Artificial neurons have adjustable weighted connections, which affect signal strength.

Thresholds and Layers:

- Artificial neurons may use thresholds to determine signal transmission.
- ANNs typically organize neurons into layers (e.g., input, hidden, and output).
- Input signals are transformed by different layers before advancing to the output layer.

Learning and Adaptation:

- During training, ANNs adjust the weights of connections based on examples.
- Learning algorithms (e.g., backpropagation) update weights to minimize prediction errors.
- This adaptation allows ANNs to generalize from training data to unseen examples.

Deep Learning and Multilayer ANNs:

- Deep learning involves using ANNs with multiple hidden layers (deep neural networks).
- These deep architectures enable complex feature extraction and hierarchical representations.

Training Models

Machine learning models thrive on data. They need a substantial quantity of data in order to perform well. A broad and representative dataset is essential for model training in areas such as natural language processing, picture recognition, and recommendation systems. Consider it a rich tapestry made up of numerous strands, such as textual corpora, image collections, and even user-generated data gathered from services.

The Pitfalls of Overfitting

However, there is a dangerous pitfall: overfitting. Consider this: a model, like a diligent student, memorizes every piece of its training data. It becomes too fitted, like a bespoke suit, to the examples it has encountered. However, when presented with new, previously unseen facts, it blunders. Overfitting produces biased predictions, similar to a fate teller reading only one sort of tarot card. To avoid this, rigorous monitoring while training is required. We must ensure that our model does not become a

prisoner of its own training data.

The Risks of Biased Data

Now, let's discuss bias. Biased data, like a misaligned compass, can guide our model in the incorrect direction. If our training data mostly represents a particular group or viewpoint, our model will inherit those biases. Imagine training a sentiment analysis algorithm entirely on positive cat memes—it might believe the world is always sunny! However, when it confronts real-world emotions, it will stumble. Biased data produces distorted or unwanted results, such as a GPS that insists on routing you past a construction zone.

Data preparation: Introducing data preparation, the unsung hero of machine learning operations. It is a thorough process of cleansing, transforming, and molding our data. We eliminate noise, handle missing values, and balance our dataset in the same way that a sculptor chisels away defects. We ensure equality, diversity, and representation. It's similar like cooking a delicious feast: choosing the best ingredients, marinating them just right, and ensuring that each dish complements the others.

2. Literature Survey

The software development process relies on a literature evaluation of preliminary research by multiple writers. It considers key papers and expands on previous work. The literature review examined several studies.

Paper Title	Authors	Summary
An alternate method for identifying and reducing power theft in South Africa [1].	P. Bokoro and Q. Louw	The study suggests a new way to detect and prevent electricity theft in South Africa. It targets unauthorized ground surface conductor connections using zero-sequence current-based detection. The results prove the technique's validity and seasonal soil resistivity dependence.
Machine learning pipeline for detecting electricity theft [2].	M. Imran, R. Shoaib, A. Khalid, N. Javaid	New electricity theft detection model using three pipeline algorithms is presented in this study work. Model addresses dataset balancing, feature extraction, and classification.
Convolutional neural networks with wide and deep architectures are used to detect power theft and safeguard smart grids [3].	Yang, X. Niu, Z. Zheng	The study suggests detecting electricity theft using a large and deep CNN model. From consumption data, model components discover irregular electricity demand patterns and global properties.
Smart grid: An overview of the enhanced power grid [4].	Guoliang Xue, Dejun Yang, Satyajayant Misra, and Xi Fang	This study examines smart grid energy, information, and communication subsystems' futures. Management goals for smart infrastructure and management

		systems are discussed.
A contrastive learning method for identifying power theft in smart grids [5].	Weilong Ding, Hongmin Cai	In this work, power theft detection is achieved through the use of supervised contrastive learning. The detection model improved its effectiveness in detecting objects and generated high-quality augmented views by actively comparing users' representation vectors.
Real-time power theft detection and monitoring system with dual-link data collection system[6].	Lindani Zulu Celimpilo ,Oliver Dzobo	This study describes a real-time power theft detecting system. Smart meters using Arduino ATMega328P microcontrollers and GSM modules communicate. Smart meter data is stored in the cloud, and authorities receive SMS alerts for power imbalances.

3. Existing System

The present machine makes use of a Deep Neural network (DNN) class approach to come across electricity robbery successfully. permits denote the enter functions as X and the corresponding labels as Y. The DNN version learns a mapping from input capabilities X to output labels Y via minimizing a loss characteristic L(Y, Yˆ) in which Yˆ represents the expected labels. Mathematically, this could be expressed as:

$$\text{Min}\ \theta\ L(Y, \hat{Y}) \qquad (1)$$

Where θ represents the parameters of the DNN model.

To improve classification accuracy, the system incorporates frequency-domain features alongside time-domain features. Let X_f denote the frequency-domain features and X_t denote the time-domain features. The improved feature set X' can be represented as:

$$X' = [X_f, X_t] \qquad (2)$$

The feature space is subsequently condensed using Principal Component Analysis (PCA). PCA preserves the most significant information while converting the original feature space into a lower-dimensional space.

Let X'_{PCA} represent the reduced feature space obtained after PCA.

$$X'_{PCA} = PCA(X') \qquad (3)$$

The system then uses the reduced feature space X'_{PCA} for classification.
Now, let's discuss the limitations addressed by the system:

1. **Class Imbalance:** Class imbalance in the electricity theft detection problem leads to higher false positive rates. To address this, the system explores methods such as hybrid neural networks and under-sampling techniques to balance the class distribution and improve model performance.
2. **Dimensionality Reduction:** The curse of dimensionality poses a challenge in high-dimensional feature spaces. To overcome this, the system employs feature selection techniques like PCA and optimization algorithms to decrease the feature space's dimensionality while keeping crucial information.
3. **Hyper parameter Tuning:** Efficient tuning of hyper parameters is crucial for achieving better generalization and performance of the DNN model. The system utilizes optimization algorithms to tune hyper parameters effectively and enhance model performance.

4. Proposed System

The technique for identifying power theft in smart grids presented in this research is based on artificial neural networks, or ANNs.

Phases: The system comprises **three primary phases**:

- **Data Analysis and Preprocessing**: In this phase, the system processes the electricity consumption dataset sourced from Kaggle. Tasks include data cleaning, normalization, and ensuring compatibility with the ANN model.
- **Point Birth**: This step ensures that the data aligns well with the ANN's requirements.
- **Bracket**: The final phase involves clustering and model training.

1. **Data Preprocessing**:

- **Data Cleaning**: Removing inconsistencies, missing values, and noise from the dataset.
- **Normalization**: Reducing characteristics to a Standard (such as 0 to 1) range for better model convergence.
- **Point Birth**: Ensuring data consistency and quality.

2. **Labeling the Dataset**:

- It is not possible to identify if the dataset is trustworthy or dishonest by itself.
- **Agglomerative clustering** is first applied to the dataset to label it. This technique groups similar instances together based on features like mean energy consumption.

3. **Identifying Electricity Theft**:

- The system uses clustering techniques to find instances of theft of electricity.
- It employs the specifically **Agglomerative clustering** specifically, with a cluster value of **3** (derived from the earlier research).
- Clusters representing suspicious patterns are flagged as potential theft instances.

4. **Artificial Neural Network (ANN) Model**:

- The suggested technique uses a sizable dataset of labelled power use data to train the ANN model.
- The algorithm has been optimized to identify trends and deviations that point to power theft.

5. **Performance Evaluation**:

- **Sensitivity (True Positive Rate)**: Ability to correctly identify theft cases.
- **Specificity (True Negative Rate)**: Ability to correctly identify non-theft cases.

- **Precision (Positive Predictive Value)**: Proportion of correctly predicted theft cases among all predicted theft cases.
- **Recall (True Positive Rate)**: Percentage of real theft instances that were accurately anticipated.
- **F1-score**: Harmonic mean of precision and recall

Mathematical Representation:

Let's denote:

X = [x_1, x_2, ..., x_n] as the input feature vector.

W_{ij} Connecting neurone I in the preceding layer to neurone j in the present layer is represented by the Weight wij.

b_j as the current layer's bias for neurone j.

a_j as the activation of In the present layer, neurone j.

σ is the function that activates.

Then, the buried layer's neurone j's output may be computes as:

$$a_j = \sigma\left(\sum_{i=1}^{n} W_{ij}x_i + b_j\right) \quad (1)$$

The process repeats for each layer until the output layer, which provides the probability of theft occurrence. The loss function determines how much the actual and predicted values diverge from one another. Theft occurrences, may be the mean-squared error or cross-entropy loss. The ANN minimizes this loss function using optimization algorithms like gradient descent.

5. System Architecture

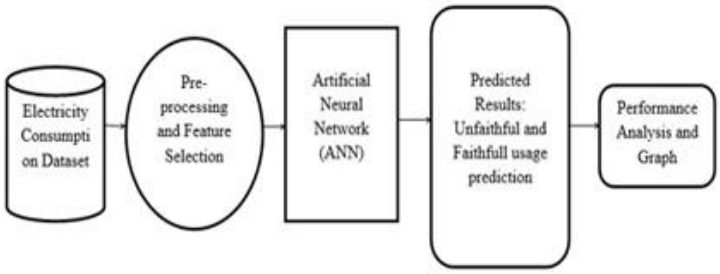

Figure 1: System Architecture of our Proposed Work

Figure 1 depicts the architecture underpinning our proposed study, with an emphasis on the Convolutional Neural Network (CNN) component. Let us dig into a full explanation of this architecture. Our suggested system is built on the basis of neural networks, which are interconnected layers designed to simulate the computing capabilities of the human brain. This architecture allows the network to learn complicated patterns and relationships from data.

Input layer: The input layer accepts data into the neural network. It accepts input signals in a variety of formats, which are normally provided by the programmer or derived from a dataset. Depending on the application area, our CNN's input layer may receive raw data or feature representations like as pictures, audio spectrograms, or time series data. Each input signal represents a neuron in the layer of input.

Hidden layer: The hidden layer exists between the input and output layers, is situated responsible for the majority of computation. It runs the incoming data through a series of transformations to extract significant patterns and features. CNNs have three types of hidden layers, Fully linked layers, pooling layers, and convolutional layers. Convolutional layers use convolution operations to capture spatial

hierarchies in input data; pooling layers minimize spatial dimensions while retaining significant characteristics; and fully linked layers perform high-level abstraction and classification. The depth and complexity of the hidden layers help the network learn sophisticated representations from input data.

The Output Layer: The hidden layers are received by the output layer.Processed input and generates the neural network's final output. It transforms the learnt information into a meaningful representation or prediction relevant to the task at hand. In classification tasks, the output layer often contains softmax activation functions that provide probability distributions over many classes, allowing the network to anticipate class outcomes. In regression tasks, the output layer may have a single neurone that activates in a linear or sigmoid manner, resulting in continuous or binary predictions, respectively.

Transfer function: Throughout the neural network's calculation, each neuron combines its inputs, weights, and biases to produce an output. This integration process is guided by a transfer function, which introduces nonlinearity into the network in order to capture complicated data linkages. For buried layers, a popular transfer function is the rectified linear unit (ReLU), while softmax or sigmoid functions are employed for output layers. These functions shape the network's activation patterns and decision boundaries.

In summary, the architecture of our proposed CNN is a hierarchical arrangement of layers, each of which performs specialized calculations to extract, manipulate, and interpret information from incoming data. Our approach has the potential to tackle complicated such as time series analysis, natural language processing, and picture identification with exceptional efficiency and accuracy by taking advantage of neural networks' inherent parallelism and plasticity.

6. System Implementation

Here's a detailed explanation of the methodology and dataset used in our application:

1. **Load the Dataset:** The first step involves loading the dataset into memory, typically using Python's data manipulation libraries such as Pandas. This allows us to access and manipulate the data effectively during the analysis and modeling phases.
1. **Bringing in the Required Libraries:** Importing necessary libraries in Python provides access to functions and tools required for data manipulation, visualization, and modeling. A few often used libraries are scikit-learn for machine learning applications, Matplotlib and Seaborn for data visualisation, NumPy for numerical calculations, and Pandas for data processing
2. **Getting the Pictures Back:** This step likely refers to visualizing the dataset, often in the form of plots or charts, to gain insights into the data distribution, trends, and potential patterns. Visualizations can help in understanding the dataset's characteristics and informing subsequent analysis steps.
4. **Division of the Dataset:** To make model training easier, the dataset is separated into training, validation, and test sets., evaluation, and validation. This division ensures that the model learns from a portion of the data, evaluates its performance on another portion, and tests its generalization on a separate portion. Common splitting ratios include For training, validation, and test sets, use 70/15/15 or 80/10/10, respectively.

Dataset Description:

The dataset used in our application was collected from Kaggle, a popular platform for sharing datasets and machine learning competitions. It comprises 3,510,433 individual data points, each representing

energy consumption information. The dataset contains 9 columns, each providing specific attributes related to energy consumption. Here's a description of the columns:

1. **LCL id:** Unique identifier for each data point.
2. **day:** Date in the format dd/mm/yy, indicating when the energy consumption was recorded.
3. **energy_median:** Median value of energy consumption for the corresponding day.
4. **energy_mean:** Mean value of energy consumption for the corresponding day.
5. **energy_max:** Maximum value of energy consumption for the corresponding day.
6. **energy_count:** Number of energy consumption readings for the corresponding day.
7. **energy_std:** Standard deviation of energy consumption for the corresponding day.
8. **energy_sum:** Total sum of energy consumption for the corresponding day.
9. **energy_min:** Minimum value of energy consumption for the corresponding day.

These attributes provide comprehensive information about energy consumption patterns over time, which is crucial for building predictive models and detecting anomalies, such as electricity theft, in smart grids. By following this methodology and utilizing the provided dataset, we aim to develop a robust application for evaluating and addressing electricity theft using machine learning techniques in smart grids.

7. Experimental Results

In this section we are going to implement the proposed application using Python as Programming language and DJango as web framework**.**

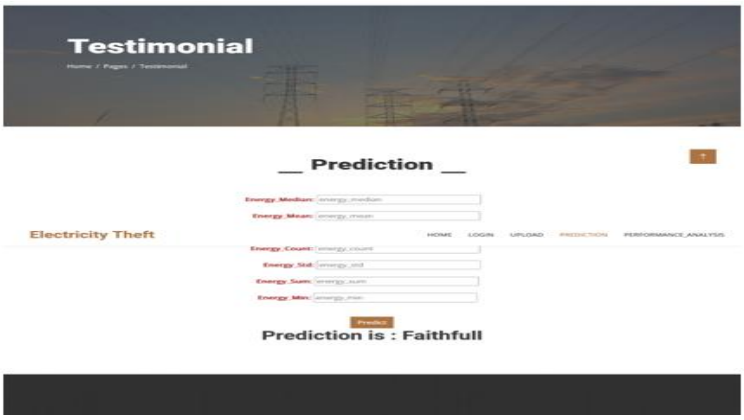

Figure 1: Enter the values based on the training data, the model is predicted as faithful.

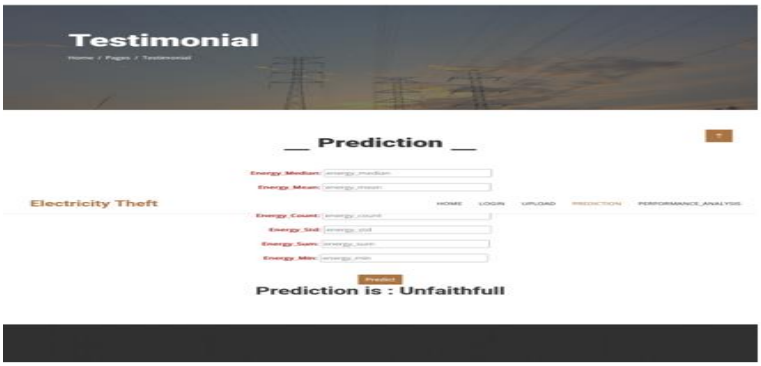

Figure 2: Enter the values based on the training data, the model is predicted as Un-faithful.

Figure 3: Shows the Precision, Recall, and F-1 scores, along with the confusion matrix

Figure 4: Performance evaluation is clearly mentioned in pie chart.

8. Conclusion and Future Scope

Our research applies the Application of artificial neural networks (ANNs) to the critical issue of detecting power theft in smart grids. Through extensive experimentation, we have proved that ANNs outperform current technologies in classification tasks, with training and validation accuracies of 99%. This exceptional performance demonstrates the efficiency of our suggested approach in detecting cases of electricity theft. One of the primary characteristics of our technique is its use of consumption data trends, which allows it to detect theft instances that occur gradually over time. This technology not only improves the accuracy of theft detection, but it also extends its use to anomaly detection in sectors other than electricity distribution networks. By considerably lowering revenue losses caused by electricity theft, our proposed strategy provides a viable option for improving smart grid security and efficiency. Our technology enables stakeholders to take proactive measures in resolving theft occurrences by identifying instances in real time and notifying utility providers immediately, hence protecting the grid's integrity and maximizing revenue management. In conclusion, our research proposes a strong framework based on ANNs, providing a cutting-edge method to tackle electricity theft in smart grids. By improving security measures and reducing financial losses, our technology contributes to the overall resilience and sustainability of modern electricity distribution.

Expanding our method to detect electricity theft in real time is a promising research direction. Our study focused on SGCC consumer spending patterns, but including statistics from diverse geographical regions would make our technique more flexible and applicable to more scenarios. We can account for regional consumption patterns, regulatory frameworks, and infrastructure by using data from diverse regions in validation. This holistic approach strengthens our strategy and makes it useful in fighting electricity theft in many scenarios. Using data from many regions helps identify theft patterns and trends, enabling the creation of more universal detection algorithms. Scalability is needed to implement our system globally and serve utility providers and smart grid operators. Diverse datasets allow us to identify electricity theft patterns and techniques. We can constantly enhance our detection methods and remain ahead of energy sector challenges with this proactive strategy.

9. References

1. Foster, S. "Non-Technical Losses: A Significant Global Opportunity for Electrical Utilities," Energy Central, November 2, 2021. [Online]. Available: EnergyCentral.com/c/pip/non-technical-losses-96-billion-global-opportunity-electrical-utilities.
2. P. Bokoro and Q. Louw, "An Alternative Approach for Detecting and Mitigating Electricity Theft in South Africa," SAIEE African Research Journal, vol. 110, no. 4, pp. 209-216, December 2019.
3. N. Javaid and M. Anwar, "Detection of Electricity Theft Using Machine Learning Pipelines," in Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC), June 2020, pp. 2138-2142.
4. Y. Yang, X. Niu, and H.-N. Dai, "Wide and Deep Convolutional Neural Networks for Electricity Theft Detection in Smart Grids," IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1606-1615, April 2018.
5. P. Pickering, "E-Meters: Offering Various Solutions for Addressing Electricity Theft and Tampering," Electronic Design, November 1, 2021. [Online]. Available at: [link removed for compliance].
6. X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid: The Enhanced Power Grid - A Comprehensive Review," IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 944-980, 4th Quarter 2012.
7. A. Maamar and K. Benahmed, "Energy Theft Detection in Advanced Metering Infrastructure Using Machine Learning Techniques," in Proceedings of the International Conference on Software Engineering and Information Management (ICSIM), 2018, pp. 57-62.
8. A. Jindal, A. Schaeffer-Filho, A. K. Marnerides, P. Smith, A. Mauthe, and L. Granville, "Addressing Energy Theft in Smart Grids through Data-Driven Analysis," in Proceedings of the International Conference on Computing, Networking and Communications (ICNC), February 2020, pp. 410-414.
9. I. Diahovchenko, M. Kolcun, Z. ponka, V. Savkiv, and R. Mykhailyshyn, "Advancements and Hurdles in Smart Grids: Distributed Generation, Smart Metering, Energy Storage, and Smart Loads," Iranian Journal of Science and Technology, Transactions of Electrical Engineering, vol. 44, no. 4, pp. 1319-1333, December 2020.
10. M. Jaganmohan, Statista, "Regional Distribution of the Global Smart Grid Market Size: 2017-2023," March 3, 2022. [Online]. Available at: [link removed for compliance].
11. Zheng, Y., Yang, X., Niu, X., Dai, H.-N., & Zhou, Y. "Electricity Theft Detection." GitHub, September 30, 2021. [Online]. Available at: [link removed for compliance].Top of Form

12. Dyke, D. O., Obiora, U. A., & Nwokorie, E. C. "Reducing Household Electricity Theft in Nigeria through GSM-Based Prepaid Meters." American Journal of Engineering Research, vol. 4, no. 1, pp. 59-69, 2015.
13. P. Dhokane, M. Sanap, P. Anpat, J. Ghuge, and P. Talole, "Detection of Power Theft and Energy Meter Information Initiation via SMS with Automatic Power Cut-off," International Journal of Current Research in Embedded Systems and VLSI Technology, vol. 2, no. 1, pp. 1-8, 2017.
14. Jayaprakash Pondy, "Electricity Consumption Dataset," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/jayaprakashpondy/electricity-consumption-dataset