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

Detecting Cyber Attacks on Renewable Distributed Generation for Future Smart Electric Meters to Prevent Electricity Theft

K Vijay Kumar[1], G Mahesh Challari[2], Priyanga Chandru[3]

[1,2]Assistant professor, Department of cse, Sree Dattha Institute of Engineering and Science
[3]Assistant professor, Department of cse, Sri Indu College of Engineering and Technology

Abstract
Electricity theft represents a pressing problem that has brought enormous fiscal losses to electric mileage companies worldwide. In the United States alone, $ 6 billion of electricity is stolen annually. Traditionally, electricity theft is committed in the consumption sphere via physical attacks that include line tapping or cadence tampering. With the advanced metering structure (AMI), smart measures are installed at the guests' demesne and regularly report the guests' consumption for monitoring and billing purposes. In this environment, vicious guests can launch cyber-attacks on smart measures to manipulate the readings in a way that reduces their electricity bill. Second, the smart grid paradigm enables guests to install renewable-grounded distributed generation (DG) units at their demesne to induce energy and send it back to the grid driver and hence make a profit.

Keywords: CNN, DNN, Random Forest Algorithm, Max polling

1. Introduction

Moment, electric power loss has become one of the most conspicuous issues affecting both conventional power grids and smart grids. The electricity losses vary from country to country. The losses in the USA, Russia, Brazil, and India were 6, 10, 16, and 18, independently, of their total energy product (3). The difference between the energy produced in one system and the metered energy delivered to the druggies is known as power loss. To determine the quantum of electricity loss, smart measures in smart grids play a prominent part.

2. Literature Review

A new data-processing algorithm to cipher the missing cases in the dataset, grounded on the original values relative to the missing data point. Likewise, in this dataset, the count of electricity theft druggies was fairly low, which could have made the model hamstrung at relating theft druggies. This class imbalance script was addressed through synthetic data generation. Eventually, the results attained indicate the proposed scheme can classify both the maturity class (normal druggies) and the nonage class (electricity theft druggies) with good delicacy (5). One fashion is the maximum information measure (MIC), which can find the correlations between the nontechnical loss and certain electricity gets Of the consumer. MIC can be used to describe thefts that appear normal in shape precisely. The other fashion is the clustering fashion by fast hunt and finding viscosity peaks (CFSFDP). CFSFDP finds abnormal

druggies among thousands of cargo biographies, making it relatively suitable for detecting electricity thefts with arbitrary shapes. Next, a frame for combining the advantages of the two ways is proposed. Numerical trials on the Irish smart cadence dataset are conducted to show the good performance of the concerted system (4).

3. Existing System

A deep neural network is simplest, a neural network with some position of complexity, generally at least two layers, qualifies as a deep neural network (DNN), or deep net for short. Deep nets process data in complex ways by employing sophisticated calculation modeling. To truly understand deep neural networks, still, it's still stylish to see it as an elaboration. Many particulars had to be erected before deep nets were. First, machine literacy had to be developed. ML is a frame to automate (through algorithms) statistical models, like a direct retrogression model, to get better at making prognostications. A model is a single model that makes prognostications about a commodity. Those prognostications are made with some delicacy. A model that learns — machine literacy — takes all its bad prognostications and tweaks the weights inside the model to produce a model that makes smaller miscalculations. Deep neural nets, also, subsidize the ANN element. They say, if that works so well at perfecting a model — because each knot in the retired subcase makes both associations and grades the significance of the input to determining the affair — also why not mound further and further of these upon each other and benefit indeed more from the retired subcase. So, the deep net has multiple retired layers. 'Deep' refers to a model's layers being multiple layers deep.

4. Proposed System

This dataset contains information on the amount of electricity each consumer used. Columns contain the dates and Rows refer to the consumers. This dataset contains the electricity consumption for the year 2021.

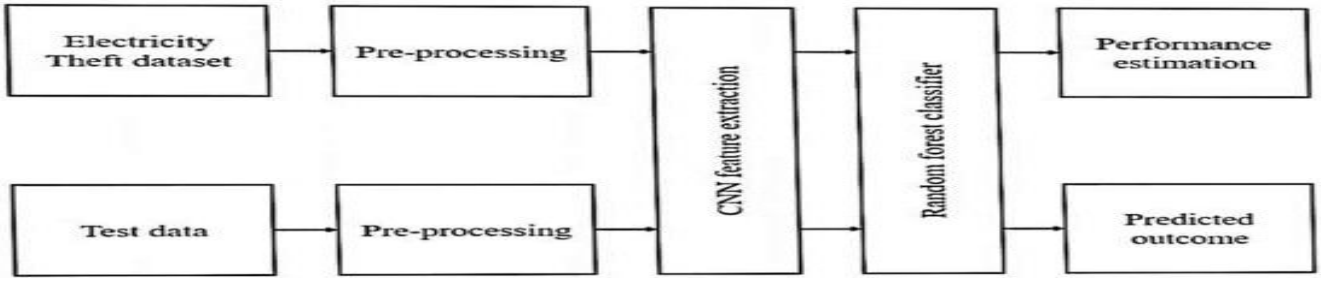


Fig 4.1: Block Diagram of Proposed System

Step 1 Data pre-processing Data processing is a process of preparing the raw data and making it suitable for a machine literacy model. It's the first and pivotal step in creating a machine-literacy model. When creating a machine literacy design, it isn't always the case that we come through clean and formatted data. And while doing any operation with data, it's obligatory to clean it and put it in a formatted way. So, for this, we use a data-processing task

Step 2 Unyoking the Dataset into the Training set and Test set the integrated data is subordinated to pre-processing, which includes data cleaning, normalization, and handling missing values. Data quality and integrity are assured to grease accurate ML model training**.**

Step 3 Training sets a subset of the dataset to train the machine literacy model, and we formerly know the affair.

Step 4 Test set a subset of the dataset to test the machine literacy model, and by using the test set, the model predicts the affair.

Step 5 Max Pooling Layer This subcase mitigates the number of parameters when there are larger size images. This can be called subsampling or down slice which mitigates the dimensionality of every point chart by conserving the important information. Max pooling considers the maximum element from the remedied point chart.

4.1: Proposed Algorithm Steps

Random Forest Algorithm

The Random Forest algorithm is a well-known method in machine learning falling under the category of supervised learning. It operates on the principle of ensemble learning, a technique that involves combining multiple classifiers to address complex problems and enhance model performance. In line with its name, Random Forest is a classifier comprising numerous decision trees built on diverse subsets of the provided dataset. The algorithm then averages the outcomes from these trees to improve the predictive accuracy of the dataset.

In contrast to relying on a single decision tree, the Random Forest extracts predictions from each tree and makes its final prediction based on the majority votes of these forecasts (**see Figure 4.1**). The use of a greater number of trees in the forest contributes to heightened accuracy and helps mitigate the risk of overfitting.

Step 1 involves selecting n random records from a dataset containing k records in the Random Forest process.

Step 2 entails constructing individual decision trees for each of the selected samples.

Step 3 results in the generation of an output by each decision tree.

Step 4 determines the final output by employing Majority Voting for classification or Averaging for regression.

Important Features of Random Forest:

Maintaining diversity is ensured as not all attributes, variables, or features are considered when constructing an individual tree; each tree possesses its own unique characteristics. The Random Forest methodology demonstrates resilience against the curse of dimensionality because each tree does not encompass all features, resulting in a reduction of the feature space. Moreover, the creation of parallelized trees is conducted independently, utilizing distinct data and attributes. This independent approach enables efficient CPU utilization in the construction of random forests, further contributing to a reduction in the feature space.

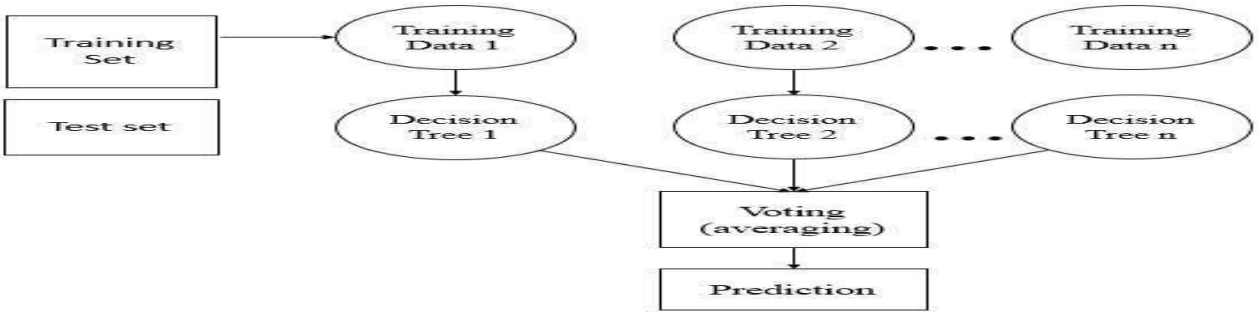


Fig.4.1: Random Forest algorithm

In a random forest there is no need to separate the data into training and testing sets since 30% of the data remains unseen by the decision tree, ensuring a built-in train-test split.
Stability is achieved through the reliance on majority voting or averaging, contributing to consistent and reliable results.

4.2 Feasibility Study:
4.2.1. Comprehensive Libraries: Python comes with an extensive library that includes code for various purposes such as regular expressions, documentation generation, unit testing, web browsers, threading, databases, CGI, email, image manipulation, and more. This eliminates the need to manually write complete code for these functionalities. Additionally, Python's versatility allows code to be extended to other languages, offering flexibility, particularly in projects.
4.2.2. Embed ability: In addition to its extensibility, Python is embeddable. It allows you to integrate Python code into the source code of another language, like C++. This capability enables the addition of scripting features to code in a different language.
4.2.3. Enhanced Productivity: Python's simplicity and rich libraries contribute to increased programmer productivity compared to languages such as Java and C++. The language's requirement for less code to achieve more tasks further enhances efficiency.
4.2.4. IoT Opportunities: Python serves as the foundation for emerging platforms like Raspberry Pi, positioning itself favorably for the future of the Internet of Things. This connection with real-world applications enhances Python's relevance in the rapidly evolving technological landscape.
4.2.5. Simplicity and Ease: Unlike Java, where creating a class is necessary for a simple 'Hello World' printout, Python accomplishes the same with just a print statement. Python is known for its simplicity, ease of learning, understanding, and coding, making it a preferred choice for newcomers. This simplicity can be challenging to those accustomed to more verbose languages like Java.
4.2.6. Readability: Python's less verbose syntax makes code reading akin to reading English. The absence of curly braces to define blocks, coupled with mandatory indentation, contributes to code readability. This readability factor facilitates the learning and comprehension of Python code.
4.2.7. Object-Oriented: Python supports both procedural and object-oriented programming paradigms. Functions aid in code reusability, while classes and objects enable the modeling of real-world entities. The encapsulation of data and functions within a class enhances the organization of code**.**

4.3. Sequence Diagram:
A sequence diagram illustrates the interaction among various objects within the system. One crucial feature of a sequence diagram is its time-ordered nature, signifying the detailed depiction of the sequence of interactions between objects in a step-by-step manner

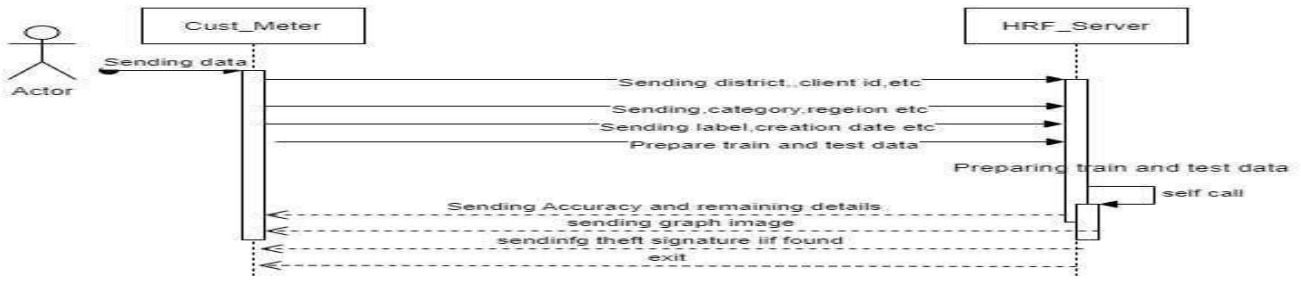


Fig 4.3.1: Sequence Diagram

5. Results

A sequence diagram illustrates the interaction among various objects within the system. One crucial feature of a sequence diagram is its time-ordered nature, signifying the detailed depiction of the sequence of interactions between objects in a step-by-step manner

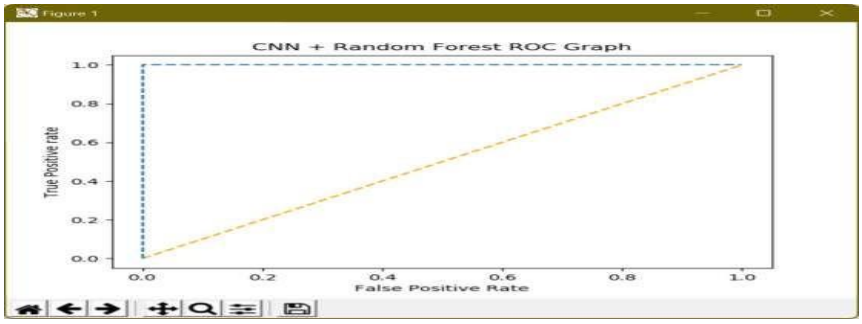


Fig 4.4.1 CNN + Random Forest ROC Graph

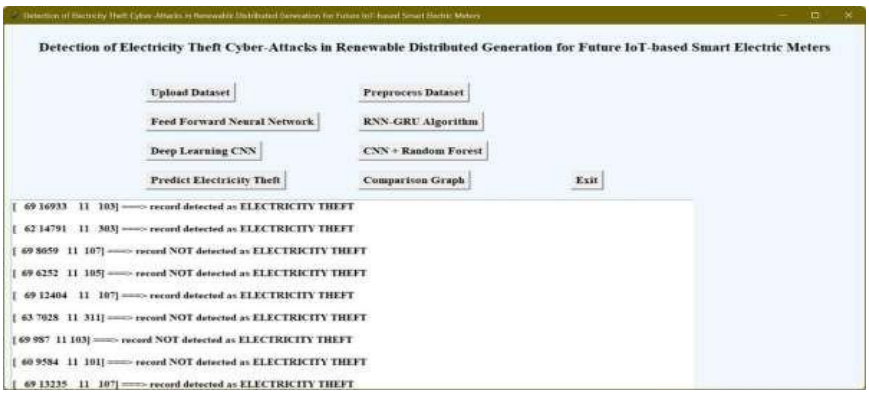
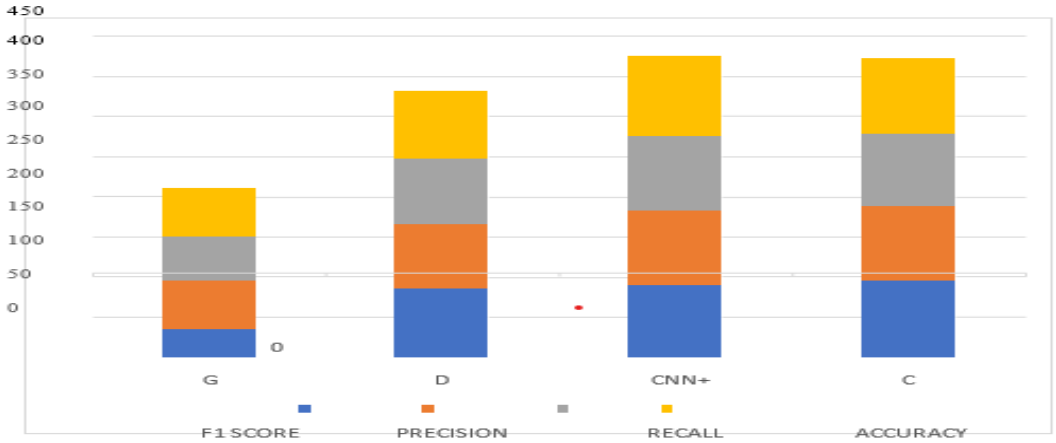
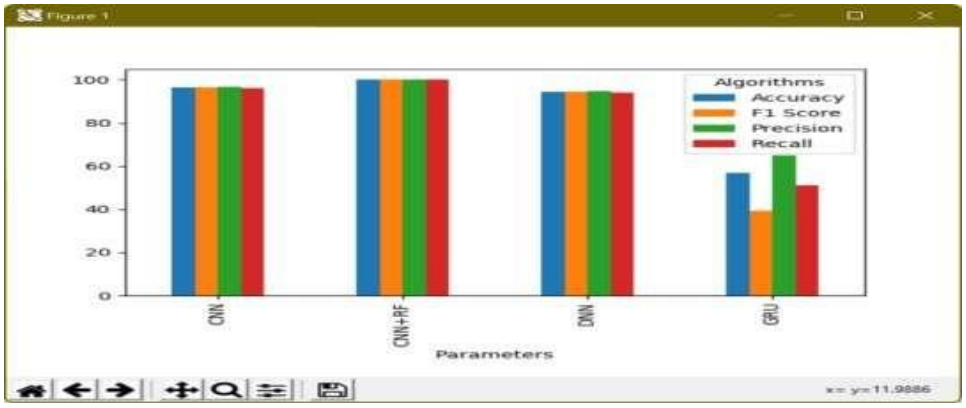


Fig 4.4.2 Predicting Electricity Theft

In the provided interface, within the square brackets, there is visible information labeled as TEST data. Following the arrow =□ symbol, there is an indication of THEFT detection with the outcome 'THEFT NOT DETECTED.' To view the corresponding graph, click on the 'Comparison Graph' button. The resulting graph displays the x-axis representing algorithm names, with each distinct color bar denoting various metrics such as accuracy, precision, recall, and FSCORE. The y-axis reflects score values. Notably, across all algorithms, Hybrid Random Forest exhibits superior performance.



Result Comparison Graph 1



Result Comparison Graph 2

6. Conclusion

The escalating global energy crises demand increased attention to both expanding energy production and conservation efforts. Electricity generation occurs through various methods, and the synchronized delivery to a central grid facilitates usage. Losses, whether technical or non-technical, pose challenges. Technical losses, discussed in the mathematical modeling section, can be calculated abstractly. Non-technical losses, including electricity theft, can be assessed when technical losses are known. Mitigating theft is crucial for preserving economic resources. Smart meters emerge as a robust solution for minimizing electricity theft due to their high security, efficiency, and resilience against various theft techniques seen in electromechanical meters.

In this study, the primary focus revolves around addressing theft issues. Consequently, the project assesses the performance of several deep learning algorithms, including the deep feed-forward neural network (DNN), recurrent neural network with gated recurrent unit (RNN-GRU), and convolutional neural network (CNN), specifically for detecting cyber-attacks on electricity systems.

7. Future Scope

The prospective advancements in detecting electricity theft and cyber-attacks within renewable distribution systems through IoT-based smart electric meters show great promise. With the ongoing expansion of renewable energy sources and smart grids, safeguarding the security and integrity of the energy distribution network becomes imperative. The following aspects underscore the potential future developments in this domain.

References

1. Das, A.; McFarlane, A. Non-linear dynamics of electric power losses, electricity consumption, and GDP in Jamaica. Energy Econ. 2019, 84, 104530.
2. Bashkari, S.; Sami, A.; Rastegar, M. Outage Cause Detection in Power Distribution Systems based on Data Mining. IEEE Trans. Ind. Inf. 2020.
3. Bank, T.W. Electric Power Transmission and Distribution Losses (% of output); IEA: Paris, France, 2016.
4. Zheng, Z.; Yang, Y.; Niu, X.; Dai, H.-N.; Zhou, Y. Wide and deep convolutional neural networksfor electricity-theft detection to secure smart grids. IEEE Trans. Ind. Inform. 2018, 14, 1606– 1615.
5. Hasan, M.N., Toma, R.N., Nahid, A.A., Islam, M.M. and Kim, J.M., 2019. Electricity theft detection in smart grid systems: A CNN-LSTM based approach. Energies, 12(17), p.3310.

6. K. Zheng, Q. Chen, Y. Wang, C. Kang and Q. Xia, "A Novel Combined Data-Driven Approach for Electricity Theft Detection," in IEEE Transactions on Industrial Informatics, vol. 15, no. 3, pp. 1809-1819, March 2019, doi: 10.1109/TII.2018.2873814.
7. Li, S., Han, Y., Yao, X., Yingchen, S., Wang, J. and Zhao, Q., 2019. Electricity theft detection inpower grids with deep learning and random forests. Journal of Electrical and Computer Engineering, 2019.
8. M. Nabil, M. Ismail, M. M. E. A. Mahmoud, W. Alasmary and E. Serpedin, "PPETD: PrivacyPreserving Electricity Theft Detection Scheme with Load Monitoring and Billing for AMI Networks," in IEEE Access, vol. 7, pp. 96334-96348, 2019 doi: 10.1109/ACCESS.2019.2925322.
9. Khan, Z.A., Adil, M., Javaid, N., Saqib, M.N., Shafiq, M. and Choi, J.G., 2020. Electricity theftdetection using supervised learning techniques on smart meter data. Sustainability, 12(19), p.8023.
10. Kocaman, B., Tümen, V. Detection of electricity theft using data processing and LSTM method in distribution systems. Sādhanā 45, 286 (2020). https://doi.org/10.1007/s12046- 02001512-0