A Lightweight Intrusion Detection and Electricity Theft Detection System for Smart Grid

Ayush Sinha, Ashutosh Kaushik, Ranjana Vyas, and O. P. Vyas

Abstract Smart grid systems have improved networking for power systems and many other industrial systems, but they still have many vulnerabilities, making them an easy target for cyber attacks. Recently, the number of attacks has also increased. The present work investigates the reliability and security of Smart Grid (SG). The reliability and security are investigated in two aspects that are electricity fraud detection followed by the intrusion detection system. This work presents the lightweight Intrusion detection system for SCADA and Modbus-based control systems that can detect intrusion with very high accuracy. The IDS developed is based on the ICS (industrial control system) dataset, which has 20 features (column) and 2,74,628 rows. The IDS dataset contains the Modbus packet's attributes and network and physical infrastructure attributes. The IDS work is followed by detecting electricity theft on a realistic electricity consumption dataset released by the State Grid Corporation of China. A total of 42,372 users' power usage data from 1,035 days is included in the data collection (from 1 January 2014 to 31 October 2016). Eight classifiers, as well as two basic neural networks (1DCNN and ANN), have been investigated on this dataset.

Keywords Intrusion detection · SCADA · Modbus · Electricity theft detection

Supported by C3iHub-IIT Kanpur, India.

A. Sinha (✉) · A. Kaushik · R. Vyas · O. P. Vyas
Indian Institute of Information Technology, Allahabad, Uttar Pradesh, India
e-mail: pro.ayush@iiita.ac.in

A. Kaushik
e-mail: icm2017002@iiita.ac.in

R. Vyas
e-mail: rvyas@iiita.ac.in

O. P. Vyas
e-mail: opvyas@iiita.ac.in

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024

S. J. Patel et al. (eds.), *Information Security, Privacy and Digital Forensics*,
Lecture Notes in Electrical Engineering 1075,
https://doi.org/10.1007/978-981-99-5091-1_5

1 Introduction

Two-way digital communication is the foundation of a smart grid, which uses digital technologies to provide power to users. Smart meters were used as part of the smart grid to help it overcome the shortcomings of traditional electrical networks. To combat climate change, improve disaster preparedness, and achieve energy independence, several governments throughout the globe are promoting the implementation of smart grids. So, two-way communication is being used to govern the usage of appliances in smart grid technology. However, the widespread availability of Internet connectivity has made the smart grid more viable to deploy. Users, operators, and automated systems can swiftly adapt to changes in smart grid conditions thanks to the efficient transmission of information through a wide range of smart grid devices.

SCADA systems are used in industrial smart grid infrastructure. Supervisory control and data acquisition (SCADA) is a group of software tools used to monitor, control, and collect data from industrial processes in real time from various distant locations. Data-driven choices about an organization's industrial operations are made possible by SCADA. Hardware and software components are both included in SCADA systems. Data is collected and transferred to field controller systems, which send it to other systems for processing and presenting to an HMI in real time. SCADA systems also keep track of and report on all process occurrences. Alarms are sounded in SCADA applications when dangerous situations arise. Mostly, SCADA uses the Modbus protocol for communication and managing SG. A serial communication protocol designed by Modicon for use with their programmable logic controllers, Modbus was released by Modicon in 1979. It is a way of sending data between serial-connected electrical equipment. It is termed a Modbus Master and a Modbus Slave when a device requests information from another. Every Slave in the 247-slave Modbus network has its Slave Address ranging from 1 to 247. It is also possible for the Master to transmit data to the Slaves. Intrusion detection is being done on the dataset, which consists of packets of Modbus.

2 Literature Review

Authors in their proposed approach in [4] used the temporal behavior of frequently occurring patterns in the SCADA protocols to identify assaults on SCADA systems using an Intrusion Detection System (IDS) specialized to SCADA. When it detects aberrant activity, the IDS sounds an alert. The IDS detected a significant number of assaults, but false alarms were kept to an absolute minimum. An operating system (OS) diversity-based intrusion detection system for SCADA systems is presented in this [5] research as a new and reliable intrusion detection method. SCADA communication over time is analyzed at the OS level, and the most suited OS is selected for intrusion detection based on reliability. According to experiments, OS diversity gives a wider range of intrusion detection options, increasing detection accuracy by

up to eight additional attack types. As a result of their idea, the system's accuracy can be improved by up to 8% on average when compared to a single OS method in the best situation. Anomaly detection systems (AbIDS) may be used to identify a stealthy cyber assault on the SCADA control system, which is being researched in this [6] work. Intending to choose a more effective IDS for SCADA security, we used the IDS tools Snort and Bro throughout the design phase and evaluated their detection rates and delay in alert packets. The timing-based rule is used to detect malicious packets based on the high temporal frequency of malicious packets in network traffic. They used the SCADA-based protection mechanism to shield the system from disruptions during the case study. The SCADA controller was hacked first, and then the data integrity of the system generator was compromised. Impact analysis and performance assessment of IDS tools are then carried out. A variety of network packet sizes were tested to see how quickly IDS solutions could detect cyber-attacks, and the findings showed that they were. Data from a gas pipeline system given by Mississippi State University is used in this [7] research to evaluate the effectiveness of Machine Learning (ML) in detecting intrusions in SCADA systems (MSU). This work makes two contributions: Two methods of data normalization were evaluated, one for accuracy and precision, and the other for recall and F1-score for intrusion detection, for a total of four methods of missing data estimates and normalization. There are two types of classifications distinguished here: binary and categorical. This research shows that RF has a high F1-score of 99% for detecting intrusions. Four distinct CPS datasets, this [8] research compares the performance of several machine learning techniques. To begin, the accuracy, precision, recall, F1-score, and AUC of machine learning algorithms are all measured and evaluated. It is also important to keep track of the amount of computing needed for training, prediction, and deployment. For critical infrastructure with diverse computing and communication limits, our extensive experimental findings will assist in choosing the appropriate machine model. According to the results of the experiments, a linear model is quicker and more suited for CPS bulk prediction. The decision tree is a suitable model for detection performance and model size.

This [9] research employs a SCADA dataset including DoS assaults and running the IEC 60870-5-104 protocol. The protocol will be wrapped into TCP/IP before being transferred so that the treatment in detecting DoS attacks in SCADA networks utilizing the IEC 104 protocol is not significantly different from a regular computer network. Intrusion detection systems (IDSs) are used to identify DoS attacks on the SCADA network using three machine learning approaches: Decision Tree, Support Vector Machine, and Gaussian Nave Bayes. 99.99 percent of the time, tests on the testing and training datasets reveal that the decision tree technique has the best performance detection. A deep learning-based intrusion detection system for SCADA networks is proposed in this [10] study to defend ICSs against conventional and SCADA-specialized network-based assaults. To define significant temporal patterns of SCADA data and identify periods when network assaults are occurring, we suggest using a convolutional neural network (CNN) rather than hand-crafted characteristics for individual network packets or flows. In addition, we devise a re-training method that allows SCADA system operators to augment our neural network models using

site-specific network attack traces. A deep learning-based solution to network intrusion detection in SCADA systems was shown to be effective in our tests utilizing actual SCADA traffic datasets, with high detection accuracy and the capacity to manage newly discovered threats. Using the autoencoder deep learning model (AE-IDS), we create an IDS for the SCADA system in this [11] study. The most often used SCADA communication protocol in the power substation is DNP3, which is the objective of the detection model. SCADA systems are particularly vulnerable to data injection and modification assaults, which fall under the broad category of "cyberattacks". This research presents the training of an autoencoder network using 17 data characteristics collected from DNP3 transmission. We examine the accuracy and loss of detection of several supervised deep learning algorithms by measuring and comparing the results. Other deep learning IDS models perform better than the unsupervised AE-IDS model.

3 Problem Definition

1. The first primary objective of this work was to build a highly accurate intrusion detection system based on physical and network parameters for MODBUS-based systems while reducing the intrusion detection algorithm's reliance on domain knowledge, i.e., the algorithm should not be pre-fed background information.
2. Another problem in smart grid infrastructure is electricity theft. The second objective of this study was to design the system to detect the same. Various ML techniques and classifiers are deployed and experimented with improving test accuracy.

4 Dataset and Proposed Methodology

In this section, we will discuss both datasets, dataset features, processing, and other details. We will also see the proposed methodology and what we plan to solve the problem at hand. In the next section, we will see the results of the methodology.

4.1 Intrusion Detection System

Dataset ICS(industrial control system: The system is simply a gas pipeline system that relays information back to SCADA about various system characteristics. Using the MODBUS packet, we can quickly determine the physical parameter's value (for example, pressure). Now, the SCADA may give control instructions based on these data. SCADA, for example, receives a value of X-Y kPa from the field while the

```
% Mississippi State SCADA Lab
% Gas Pipeline Dataset
%
% Sources:
%       Author: Ian Turnipseed
%       Advisor: Dr.Morris
%       Date: Fri Dec 19 09:53:19 2014
%
@relation gas

@attribute 'address' real
@attribute 'function' real
@attribute 'length' real
@attribute 'setpoint' real
@attribute 'gain' real
@attribute 'reset rate' real
@attribute 'deadband' real
@attribute 'cycle time' real
@attribute 'rate' real
@attribute 'system mode' real
@attribute 'control scheme' real
@attribute 'pump' real
@attribute 'solenoid' real
@attribute 'pressure measurement' real
@attribute 'crc rate' real
@attribute 'command response' {0,1}
@attribute 'time' real
@attribute 'binary result' {'0','1'}
@attribute 'categorized result' {'0','1','2','3','4','5','6','7'}
```

Fig. 1 Attributes in dataset

pipeline pressure should be X kPa. SCADA then sends an order to raise the pressure by Y percent. The following is the data that the Modbus packet conveys (Fig. 1).

Proposed Methodology—IDS The objectives of algorithm selection are as follows:

1. If it can tell the difference between fault and assault, it is doing its job correctly. It should also be able to tell what kind of assault it is.
2. The algorithm must be lightweight: it should not interfere with the core function of the industrial computer.
3. No previous domain knowledge should be required: no networking or gas pipeline data should be provided in this case.

Because of reason number two, we decided against using a neural network. Logistic regression was our first thought when attempting to determine the likelihood of an assault. There was no noticeable difference in accuracy when the value of "C" was altered. Also, the next natural step was to explore SVM and try different improvisations.

4.2 Electricity Theft Detection

Dataset The State Grid Corporation of China made this data public (SGCC). A total of 42,372 power users were tracked for 1,034 days (1 Jan 2014–31 Oct 2016). One individual out of the first 3615 has been tagged as a fraudster. There are 40258 clients listed in the actual data.

Data Preprocessing Missing values are common in electricity usage statistics. The failure of smart meters, the inconsistent transfer of measurement data, the unannounced system maintenance, and storage concerns are all contributing factors. To fill in the blanks, we'll use the interpolation approach using the equation below:

$$f(x_i) = \begin{cases} \frac{x_{i-1}+x_{i+1}}{2} & x_i \in \text{NaN}, x_{i-1}, x_{i+1} \notin \text{NaN} \\ 0 & x_i \in \text{NaN}, x_{i-1} \ or \ x_{i+1} \in \text{NaN} \\ x_i & x_i \notin \text{NaN}, \end{cases}$$

If xi is a non-numeric character or a null value in the electrical consumption statistics throughout a period, we display it as NaN. (NaN is a set). Some of the values in the data are incorrect. Here's how we get back the original value:

$$f(x_i) = \begin{cases} \text{avg}(\mathbf{x}) + 2 \cdot \text{std}(\mathbf{x}) & \text{if } x_i > \text{avg}(\mathbf{x}) + 2 \cdot \text{std}(\mathbf{x}), \\ x_i & \text{otherwise}, \end{cases}$$

Avg(x), std(x): The average value of x and the standard deviation of x are shown in this equation. Because each user's power usage always exceeds zero, we only take into account the positive deviation in the preceding calculation. We must normalize the dataset since neural networks are sensitive to a wide range of data. As for scaling, we utilized MAX-MIN scaling using the equation below:

$$f(x_i) = \frac{x_i - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}$$

Min(x) is the lowest value in x and max(x) is the highest value in x.

4.3 Data Visualization

After creating a new dataset that includes the first three rows of the original dataset and the final two rows of the original dataset (consumers without fraud), we can begin our Visualization.

We must plot customer data based on the following criterion: customers with fraud and customers without fraud. Dated customers are consumed. Consumption

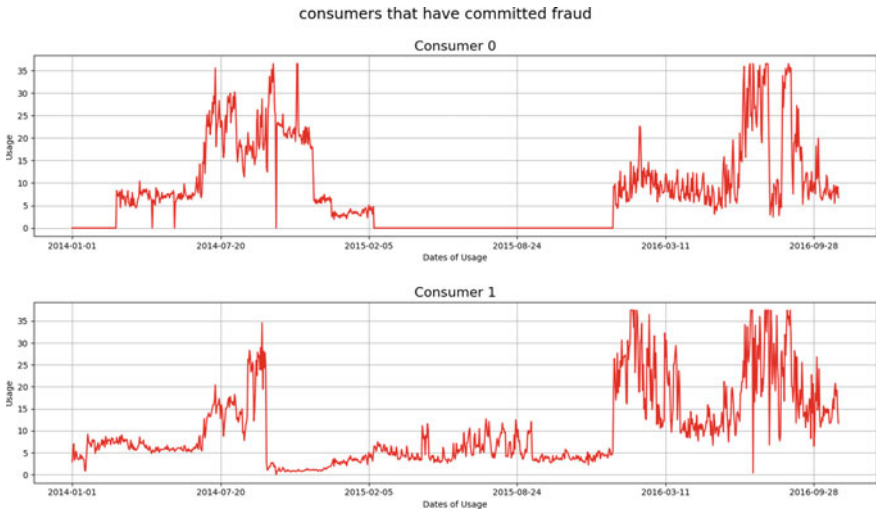

Fig. 2 Consumers who did not do fraud

data in the form of a histogram. The amount of power used in a certain area. Other data include the 50% value, maximum, and lowest value.

The first half of the dataset contains users who have committed the fraud, whereas the second half of the dataset contains the consumers who have committed the fraud. Figure 2 shows the electricity usage of the first two uses of the dataset over the whole time range. Figure 3 plots the usage of the last two users (Users 40255 and 40256) over

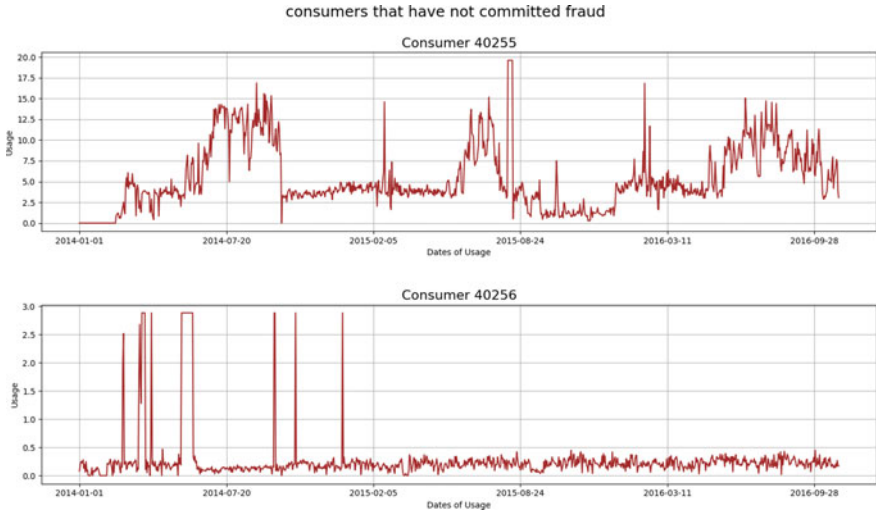

Fig. 3 Consumers who did fraud

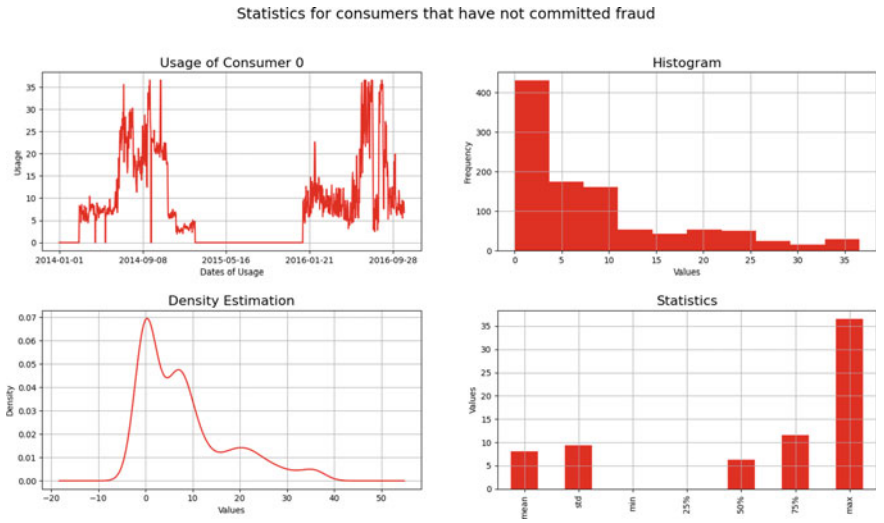

Fig. 4 Consumers who did not do fraud

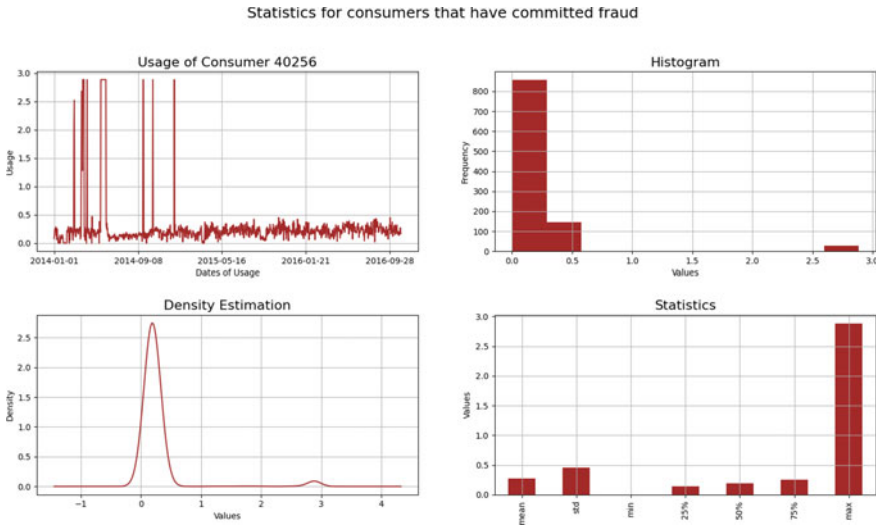

Fig. 5 Consumers who did fraud

the same period. Finally, Figs. 4 and 5 contain the shoes various statistical analyses of first and last users, and the histogram is of frequency of usage vs the amount of usage. Here, first user is a fraud, and the last user is not a fraud (Tables 1 and 2).

Proposed Methodology—ETD After processing the dataset, we now try different categories of algorithms and models and see if we can tune the parameters and get the improvement. In this existing work [14] (refer to Table 3), they already used deep

Table 1 Accuracy of basic models experimented

Model	Accuracy(%)
ANN	88.5678768157959
CNN1D	88.94066480273378
SVM	89.12705809257533
RF	89.12705809257533
DT	81.85771978875427
LR	85.425937556588354

Table 2 Accuracy of all of the classifiers experimented

Classifier	Accuracy(%)
XGB classifier	89.25132028580305
LGBM classifier	89.22025473749612
Gradient boosting classifier	89.22025473749612
CatBoost classifier	**90.59304131717925**

Table 3 Comparison of our results with the results of paper [14]

Classifier	Accuracy(%)	Accuracy achieved [14]
LR	89.25132028580305	0.8670
SVM	89.22025473749612	0.7536
RF	89.22025473749612	0.8864

neural networks and achieved excellent results. Still, we did try ANN and 1D-CNN with machine learning classifiers and wanted to improve performance in the classifier category, where we got little improvement. In the next section, let us discuss each experiment and model we tried in detail for both problems.

5 Results and Comparison

In this section, we will discuss the experiments we performed for the problem statement that we discussed and their result and compared them with the existing work.

5.1 Intrusion Detection System

Experiment 1: Logistic Regression We wanted to define the boundary of classification to be very precise and check whether having a loose or tight margin would

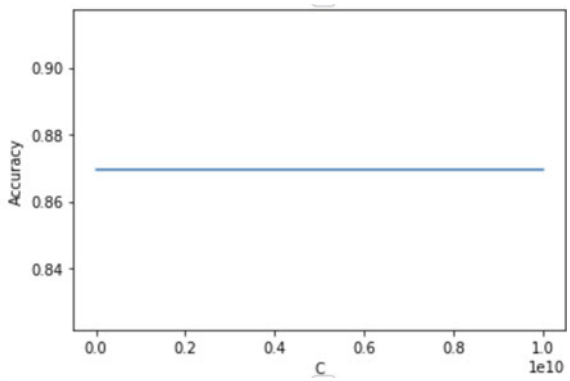

Fig. 6 Exp-1:Accuracy versus Control error V1

help us perform better, so we performed a loop on control error (C) to see whether LR's accuracy improved was still 86.94 (Refer to Fig. 6).

Experiment 2: Division of Dataset Now, One thing to notice is dataset contains both command requests and command responses. Now allowing the algorithm to differentiate between this part does not fit our third aim, but if it significantly helps the algorithm, then it is just the knowledge of request and response. So we divided the dataset into one which contains "command_request" 1 and other which contains "command_response" 2. This ultimately helped the algorithm to distinguish between request and response. Now also note that the number of responses and requests were equal, so the total number of rows after response removal is 274628/2 = 137314. This is the time it improved from 86.94 to 90.7, although when we tried to vary the value of control error (C) accuracy didn't change much. It varies from 90.3 to 90.7 for a large range of C (Refer to Fig. 7).

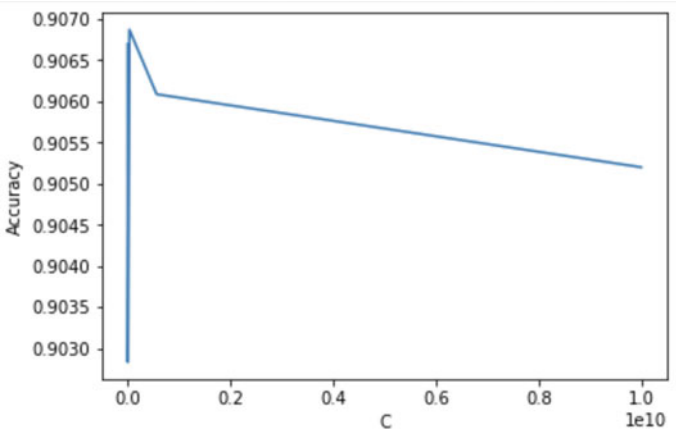

Fig. 7 Exp-2:Accuracy versus Control error V2

Experiments 3 and 4: SVM Now with SVM, we achieved an accuracy of around 94% (on the dataset without division). We tried SVM on the divided dataset and ran nested loop over C and gamma, and for C = 5 and gamma = 0.09, we got the accuracy of **99.4158%**. Another experiment with a different kernel in SVM (default is a radial basis function kernel (RBF)), i.e., polynomial kernel, resulted in reduced accuracy of 92.133. This paper [7] used SVM on the same dataset and got the best accuracy of 94.36%.

5.2 Electricity Theft Detection

Experiment 5 We have already discussed the prepossessing. Let us split the dataset into 80–20 for training and testing. Following is the table for results for every model we tested on. Now, we tried the gradient boosting method, which combines various methods which are weak and assigns weight to them. The classifier we tried vs accuracy is shown in the following Table 2.

We got 90.59 percent test accuracy with the CatBoost classifier, which is an improvement among the category of classifiers. Better results have been produced using neural networks and other combinations. However, in the case of just classifiers, this is better than the existing ones (published in the category of classifiers without using neural networks.

It is possible to increase the performance of a machine learning model based on gradient boosting and decision trees using the CatBoost Model. This may be used for categorical and continuous data values, making it even more versatile. CatBoost Classifier eases our burden of translating categorical data into the numeric form and begins creating the model, as well as we dive into the categorical values. The categorical characteristics or variables are enabled and handled automatically and treated as such. It has given us the best results as of now. Note that there is no work that compares existing methods or pre-built models (Gradient boosting versions) like we have used here.

6 Conclusion and Future Scope

In this research work, we have presented a lightweight IDS and electricity theft detection which can detect attacks with very high accuracy. We were able to get the improvement from 94.3%, which is existing work, to 99.4%. We used the ICS dataset published in 2014 and made the algorithm understand the difference between request and response, which lead to this huge spike in accuracy. We also provided it with information about the command request and response, which is the knowledge about the network packets. The second section of the work consists of the electricity theft detection on data released in 2017 by SGC of China. We tried basic methods and various pre-built versions of gradient boosting to improve the performance and

presented a comparison. A total of 10 different methods were experimented with. We established that though much recent work has already explored the neural network and other ways to optimize performance. However, for pre-existing classifiers, CatBoost is the recent one, and it gave better results than other previous classifiers. Further research can be done to improvise and not have the network knowledge while training.

7 Funding

The work is partially funded by the Department of Science and Technology(DST), and C3i-Hub (Indian Institute of Technology Kanpur), India, for the Risk Averse Resilience Framework for Critical Infrastructure Security(RARCIS) project.

References

1. http://www.ece.uah.edu/thm0009/icsdatasets/IanArffDataset.arff
2. Zheng Z, Yang Y, Niu X, Dai H-N, Zhou Y (2018) Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. IEEE Trans Ind Inf 14(4):1606–1615. https://doi.org/10.1109/TII.2017.2785963
3. Cat Boost classifier: https://github.com/catboost/catboost
4. Sayegh N, Elhajj IH, Kayssi A, Chehab A (2014) SCADA intrusion detection system based on temporal behavior of frequent patterns. In: MELECON 2014—17th IEEE mediterranean electrotechnical conference, pp 432-438. https://doi.org/10.1109/MELCON.2014.6820573
5. Bulle BB, Santin AO, Viegas EK, dos Santos RR (2020) A host-based intrusion detection model based on OS diversity for SCADA. In: IECON 2020 the 46th annual conference of the IEEE industrial electronics society, pp 691–696. https://doi.org/10.1109/IECON43393.2020.9255062
6. Singh VK, Ebrahem H, Govindarasu M (2018) Security evaluation of two intrusion detection systems in smart grid SCADA environment. In: North American power symposium (NAPS), pp 1–6. https://doi.org/10.1109/NAPS.2018.8600548
7. Lopez Perez R, Adamsky F, Soua R, Engel T (2018) Machine learning for reliable network attack detection in SCADA systems. In: 2018 17th IEEE International conference on trust, security and privacy in computing and communications/12th IEEE international conference on big data science and engineering (TrustCom/BigDataSE), pp 633–638. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00094
8. Kumar A, Choi BJ (2022) Benchmarking machine learning based detection of cyber attacks for critical infrastructure. In: International conference on information networking (ICOIN), pp 24–29. https://doi.org/10.1109/ICOIN53446.2022.9687293
9. This research employs a SCADA dataset
10. Yang H, Cheng L, Chuah MC (2019) Deep-learning-based network intrusion detection for SCADA systems. In: IEEE conference on communications and network security (CNS), pp 1–7. https://doi.org/10.1109/CNS.2019.8802785
11. Altaha M, Lee JM, Aslam M, Hong S (2021) An autoencoder-based network intrusion detection system for the SCADA system
12. Zheng Z, Yang Y, Niu X, Dai H-N, Zhou Y (2018) Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids. IEEE Trans Ind Inf 14(4):1606–1615. https://doi.org/10.1109/TII.2017.2785963

13. Abdulaal MJ et al (2022) Real-time detection of false readings in smart grid AMI using deep and ensemble learning. IEEE Access 10:47541–47556. https://doi.org/10.1109/ACCESS.2022.3171262
14. Lepolesa LJ, Achari S, Cheng L (2022) Electricity theft detection in smart grids based on deep neural network. IEEE Access 10:39638–39655. https://doi.org/10.1109/ACCESS.2022.3166146
15. Alkuwari AN, Al-Kuwari S, Qaraqe M (2022) Anomaly detection in smart grids: a survey from cybersecurity perspective. In: 3rd International conference on smart grid and renewable energy (SGRE), pp 1–7. https://doi.org/10.1109/SGRE53517.2022.9774221
16. Lee J, Sun YG, Sim I, Kim SH, Kim DI, Kim JY (2022) Non-technical loss detection using deep reinforcement learning for feature cost efficiency and imbalanced dataset. IEEE Access 10:27084–27095. https://doi.org/10.1109/ACCESS.2022.3156948
17. Ullah A, Javaid N, Asif M, Javed MU, Yahaya AS (2022) AlexNet, adaboost and artificial bee colony based hybrid model for electricity theft detection in smart grids. IEEE Access 10:18681–18694. https://doi.org/10.1109/ACCESS.2022.3150016
18. Xia X, Xiao Y, Liang W, Cui J (2022) Detection methods in smart meters for electricity thefts: a survey. Proc IEEE 110(2):273–319. https://doi.org/10.1109/JPROC.2021.3139754
19. Zhao Q, Chang Z, Min G (2022) Anomaly detection and classification of household electricity data: a time window and multilayer hierarchical network approach. IEEE Internet Things J 9(5):3704–3716. https://doi.org/10.1109/JIOT.2021.3098735
20. Althobaiti A, Jindal A, Marnerides AK, Roedig U (2021) Energy theft in smart grids: a survey on data-driven attack strategies and detection methods. IEEE Access 9:159291–159312. https://doi.org/10.1109/ACCESS.2021.3131220
21. https://www.sciencedirect.com/science/article/pii/S2090447920301064
22. https://accelconf.web.cern.ch/ica99/papers/mc1i01.pdf
23. Reynders D, Mackay S, Wright E (2004) Modbus overview. Edwin PY-2004/12/31, SP-132, EP-141, SN-9780750663953, T1. https://doi.org/10.1016/B978-075066395-3/50012-7
24. Ghosh S, Dasgupta A, Swetapadma A (2019) A study on support vector machine based linear and non-linear pattern classification. In: International conference on intelligent sustainable systems (ICISS), pp 24–28. https://doi.org/10.1109/ISS1.2019.8908018
25. Huang M (2020) Theory and implementation of linear regression. In: 2020 International conference on computer vision, image and deep learning (CVIDL), pp 210–217. https://doi.org/10.1109/CVIDL51233.2020.00-99
26. Ho TK (1995) Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition, vol 1, pp 278–282. https://doi.org/10.1109/ICDAR.1995.598994
27. Navada A, Ansari AN, Patil S, Sonkamble BA (2011) Overview of use of decision tree algorithms in machine learning. IEEE Control and System Graduate Research Colloquium 2011:37–42. https://doi.org/10.1109/ICSGRC.2011.5991826
28. Uhrig RE (1995) Introduction to artificial neural networks. In: Proceedings of IECON '95—21st annual conference on IEEE industrial electronics, vol 1, pp 33–37. https://doi.org/10.1109/IECON.1995.483329
29. Upadhyay D, Manero J, Zaman M, Sampalli S (2021) Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids. IEEE Trans Netw Serv Manag 18(1):1104–1116. https://doi.org/10.1109/TNSM.2020.3032618