Exploring Ensemble Classifiers for Detecting Attacks in the Smart Grids

Kudrat Jot Kaur
School of Electrical Engineering and Computer Science
Washington State University
Pullman, USA
kudrat.kaur@wsu.edu

Adam Hahn
School of Electrical Engineering and Computer Science
Washington State University
Pullman, USA
ahahn@eecs.wsu.edu

***Abstract* – The advent of machine learning has made it a popular tool in various areas. It has also been applied in network intrusion detection. However, machine learning hasn't been sufficiently explored in the cyberphysical domains such as smart grids. This is because a lot of factors weigh in while using these tools. This paper is about intrusion detection in smart grids and how some machine learning techniques can help achieve this goal. It considers the problems of feature and classifier selection along with other data ambiguities. The goal is to apply the machine learning ensemble classifiers on the smart grid traffic and evaluate if these methods can detect anomalies in the system.**

Keywords – Smart grid security, machine learning, intrusion detection.

1. INTRODUCTION

With the increasing threats towards the critical infrastructures such as smart grids, the need to identify such anomalies and attacks has become extremely crucial. One of the rapidly emerging tools in scientific community is Machine Learning. The last decade has seen major growth in the field of machine learning and its applications. It has been used extensively in computational biology, speech recognition, self-driving vehicles, and many more fields. The scope of these tools is tremendous and they have been used to implement network security as well. However, the use of these tools for detecting attacks in smart grids is not a widely-researched topic. The motivation behind this paper is that the smart grids have traffic that can be used with the machine learning classifiers to generate successful results.

Several factors need to be understood to enhance the effectiveness of machine learning tools in intrusion detection in smart grids. These range from the dataset definition, feature selection, and labeling of the data to the selection of a classifier.

As mentioned in [1], the strength of machine learning is to find something similar to what has been previously seen. Therefore, expecting an algorithm to detect different kinds of attacks and separating them from different kinds of benign traffic becomes hard. Besides this, the requirement for the data to be labeled with the right features is an important issue that needs to be handled carefully to get the viable results.

This paper looks into the traffic from a Smart City Testbed. Since this traffic is the communication between power system devices over fixed standards, the data is consistent. The uniqueness of this traffic is that it is low in entropy as the generic communication between the devices doesn't alter. This helps shape up a definitive dataset that can be used as a worthwhile training and testing set and generate more accurate results. In order to maintain generality towards different kinds of attacks, the classification has been performed using ensemble classifiers.

The rest of the paper is divided in the following sections. Section II briefly describes the Smart City Testbed at Washington State University along with the data used. Section III looks into various challenges while using machine learning for attack detection.

Section IV discusses the evaluation of the classification process and the results obtained.

2. SMART CIY TESTBED

The Smart City Testbed at Washington State University is a state of the art testbed that incorporates the cyber-physical structure of a smart grid. It is a platform to find ways to make the smart grids more secure and robust amidst the threats that the cyber-physical systems face.

The testbed has a Distributed Management System (DMS) from GE that has been programmed to represent the DMS of the Pullman city, WA. This acts as the control center of the testbed. Besides this, the testbed has Itron smart meters which are connected to the DMS through a gateway. The meter data is sent to Itron's cloud and can be read via Itron's Openway Gateway Application.

The distribution substation model of the testbed constitutes ABB's distribution feeder relays that are connected to Circuit Breaker simulators. These are connected to the control center through the ABB COM600 interface. The communication between the COM600 and DMS is carried out through DNP3 protocol. In addition, there are two protection IEDs from Alstom and SEL in the transmission substation that communicate to the control center through IEC61850 standard. The interconnectivity of the devices in the testbed is set over NS3 network simulator. Fig 1 shows an overview of the testbed.

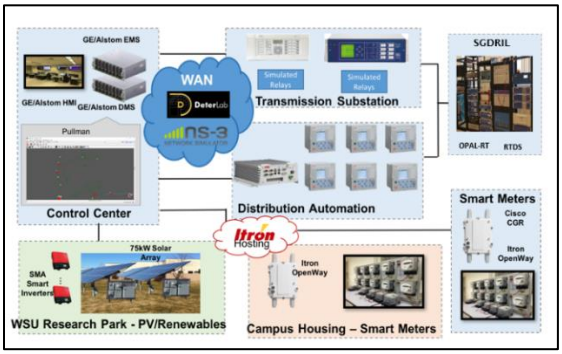

Fig 1. Smart City Testbed Architecture

The traffic in the testbed is the communication between various devices which is captured using Wireshark and ELK (Elasticsearch-Logstash-Kibana) stack. It consists, primarily, of DNP3 sessions, network support and management protocols along with HTTP sessions. The traffic is monitored over the switches which collect the Netflows. This traffic comes from the control center, the distribution substation, and other workstations in the testbed. The data has been collected over a month's period. As discussed earlier, Table I shows the consistency of the traffic. Each of the feature has a small domain. The table implies that there are a total of 12 different source and destination IP addresses throughout the dataset. This results in the low entropy of the system since the randomness between various parameters is low.

Table I. Smart Grid Traffic

Features	No. of different values
Source IP	12
Destination IP	12
Protocol	10
Source Port	24
Destination Port	21

3. CHALLENGES OF USING MACHINE LEARNING

The first challenge in using machine learning for the purpose of detecting attacks is the formation of the dataset. Majority of papers ([2], [3], and [4]) that have implemented machine learning for intrusion detection have done so on very old datasets with attacks and features that do not hold a lot of significance in present time. Besides, there aren't any smart grid datasets publicly available. Labeling the network traffic is not only a tedious task but a crucial one too. A poorly labeled dataset can't produce optimal results no matter how strong the classifier is.

Another concern is to figure out the right features for the dataset. The network traffic captures numerous amounts of features, most of which are not very helpful for process of intrusion detection. In fact, some of these might add ambiguity or bias and lead to overfitting. The features should be selected in such a way that an attack would cause those features to change and behave abnormally.

Once the dataset has been formed, the next step is to choose a classifier that can perform with the best results. Different papers have used different kinds of classifiers to achieve this task. The review [5] shows how the research has shifted towards ensemble classifiers instead of individual classifiers over the years. This is because the use of multiple weak classifiers helps in reducing the overfitting and overcoming the shortcomings of the individual classifiers. Also, in case of network traffic, which is

diverse in nature, the use of different classifiers helps in pointing out different kinds of attacks which would be difficult for a single algorithm.

The variability in the collected network traffic adds to all these problems. Data collected on day 1 might not resemble the data collected at the end of the month. This makes it harder for the classifiers to learn the datasets and evaluate the traffic.

4. EVALUATION OF ENSEMBLE CLASSIFIER

This paper considers the data collected in the Smart City Testbed from the workstation that communicates with the control center and the distribution relays. The dataset that has been created for this research keeps in mind some of the main issues mentioned earlier and overcomes those problems in hopes of using the machine learning tools to the best of their capabilities. The dataset has very low entropy and the traffic doesn't add a lot of ambiguities.

The features selected for this dataset are – source and destination addresses, source and destination ports, protocol, and the length of the packet. To check if all the features contribute to the accurate classification of the data, they were all removed one by one. The accuracy of the classifiers dropped when a feature was removed.

After the features have been selected, the data is labeled. The methods of clustering were avoided because they don't do well in case of outliers [7]. The data is labeled manually. This step might have some errors and can cause certain vagueness in the dataset especially for the packets that come from unknown sources/ports. It is assumed that the data has been labeled as correctly as possible.

The ensemble classifier Bootstrap Aggregation (BAGGING) was chosen as the learning classifier. Bagging is a meta-algorithm that aims at improving the accuracy of the basic algorithms used for classification and regression. It helps reduce the variance in the dataset. Lesser variance helps to avoid overfitting of the data. The algorithm uses the concept of model averaging.

To use bagging, a basic algorithm needs to be chosen. For this paper, the decision tree (J48) and Bayes network were chosen. Since the dataset used has very low entropy, the use of decision trees help in separating features and instances that would cause an increase in the entropy. In case of Bayes net which uses the probability distribution of the dataset, the probability distribution of the benign traffic would be higher as it constitutes majority of the data. An anomaly, on the other hand, would have a lower probability. This would help detect the outliers better. Fig 2 shows the entropy for the source IP feature of benign traffic.

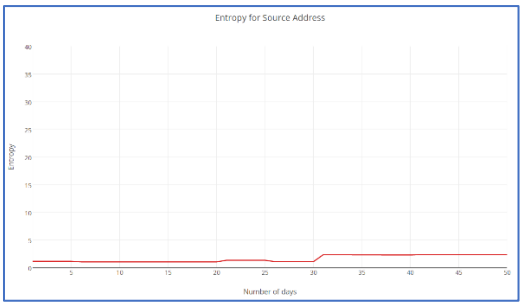

Fig 2. Entropy graph for source IP for benign traffic

The results of the classification were derived with the help of WEKA [6]. The dataset was divided into training and testing set with 20% and 80% of constitution of whole set respectively. Both the data sets include both the benign and attack traffic. Most of the data consists of benign traffic that is consistent to the testbed. Two different attacks were performed to collect anomalous traffic – VNC password attack and remote code execution attack.

The first attack aims at obtaining unauthorized access to a system. The adversary enters the VNC server and obtains the password for the control center. In the second attack, the attacker has access to the control center and sends commands to trip the circuit breakers.

These two attacks affect different kinds of features in the traffic. The VNC password attack can be recognized from the type of protocol since the normal traffic in the testbed doesn't use VNC protocol. For the remote code attack, there are out of order TCP packets that act as outliers.

When the WEKA runs the classifier, it generates confusion matrix for the dataset. A confusion matrix is the measure of how many instances have been classified correctly/incorrectly in the given dataset. It is a base to define how well a classifier performs. Fig 3 and 4 show the confusion matrix for the training (upper) and testing (lower) sets for bagging with decision tree J48 and Bayes Net respectively. This shows how many instances have been detected correctly by the classifier. Here, GOOD implies the

benign packets while BAD is the attack traffic. For instance, the bagging with J48 classified all benign packets correctly and 61 out of 66 attack packets correctly in the training set. When the classifier runs on test set, it misclassifies 2 good packets as bad and 14 bad packets as good.

	GOOD	BAD
GOOD	434	0
BAD	5	61
	GOOD	BAD
GOOD	1749	2
BAD	14	236

Fig 3. Confusion Matrix (J48)

	GOOD	BAD
GOOD	430	4
BAD	3	63
	GOOD	BAD
GOOD	1728	23
BAD	2	248

Fig 4. Confusion Matrix (Bayes Net)

While the Bayes Net seemed to have done better on the training set, J48 gives a better result on the testing set. However, the use of bagging makes the results quite impressive due to the model averaging technique and the reduction in the variance. Fig 5 and 6 show how the value of recall (number of relevant instances retrieved) for attack traffic reaches max value very soon and stays there. This implies that the classifier learns very quickly. These results may vary for different attacks.

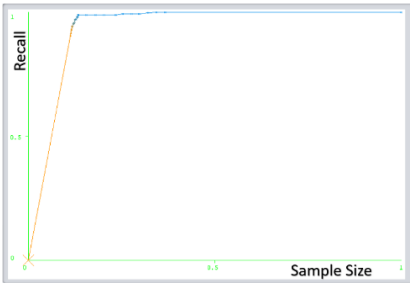

Fig 5. Attack traffic detected in Testing Set (J48)

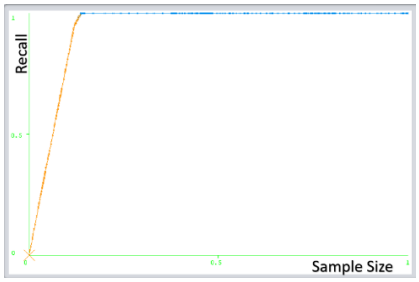

Fig 6. Attack traffic detected in Testing Set (Bayes Net)

5. CONCLUSIONS AND FUTURE WORKS

While the use of machine learning in intrusion detection for smart grids is still a field that needs more of research, these tools can be successfully implemented if the data is used appropriately. Using a traffic that has some unique properties (such as low entropy) can be used to the advantage. The feature selection and choice of classifiers is extremely crucial to get the expected results. The use of ensemble classifiers shows that great accuracy in the results can be achieved.

In future, better ways can be formulated for labeling and feature selection. It is important to have correct dataset to perform any sort of experiment. Methods such as Principal Component Analysis (PCA) can be used to reduce the dataset to features that are important for classification. Other algorithms can be tested along with bagging. The dataset can be elaborated to include various kinds of attacks that a smart grid might have to deal with. Also, data from system logs, VPN, and Windows can be studied to get an in-depth understanding of the system and find more vulnerabilities. In summation, the powerful tools of machine learning can be implemented to detect attacks in smart grids.

ACKNOWLEDGEMENT

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780.

REFERENCES

[1] R. Sommer and V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection” in Proceedings of IEEE Symposium on Security and Privacy, 2010.
[2] M. Sabhnani and G. Serpen, “Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context” in MLMTA, June 2003.
[3] H. Kayacık, A. Zincir-Heywood, and M. Heywood, “Selecting Features for Intrusion Detection: A Feature Relevance Analysis on KDD 99 Intrusion Detection Datasets” in Proceedings of the third annual conference on privacy, security and trust, 2005.
[4] Mukkamala, Srinivas, Guadalupe Janoski, and Andrew Sung. "Intrusion detection using neural networks and support vector machines." Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on Vol. 2. IEEE, 2002.
[5] C. Tsai, Y. Hsu, C. Lin, and W. Lin, “Intrusion Detection by Machine Learning: A Review” in Expert Systems with Applications, Volume 36, Issue 10, December 2009.
[6] Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.
[7] Colin Gilmore and Jason Haydaman, “Anomaly Detection and Machine Learning Methods for Network Intrusion Detection: an

Industrially Focused Literature Review", Int'l Conference Security and Management, 2016.