2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) | 978-1-6654-1502-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/SMARTGRIDCOMM51999.2021.9632

Detecting Attacks on Synchrophasor Protocol Using Machine Learning Algorithms

Kolten Knesek[1], Patrick Wlazlo[1], Hao Huang[2], Abhijeet Sahu[2], Ana Goulart[1], Kate Davis[2]
[1]Engineering Technology and Industrial Distribution, Texas A&M University
[2]Electrical and Computer Engineering, Texas A&M University
{kolten.knesek, pjrwlazlo, hao_huang, abhijeet_ntpc, goulart, katedavis}@tamu.edu

***Abstract*—Phasor measurement units (PMUs) are used in power grids across North America to measure the amplitude, phase, and frequency of an alternating voltage or current. PMU's use the IEEE C37.118 protocol to send telemetry to phasor data collectors (PDC) and human machine interface (HMI) workstations in a control center. However, the C37.118 protocol utilizes the internet protocol stack without any authentication mechanism. This means that the protocol is vulnerable to false data injection (FDI) and false command injection (FCI). In order to study different scenarios in which C37.118 protocol's integrity and confidentiality can be compromised, we created a testbed that emulates a C37.118 communication network. In this testbed we conduct FCI and FDI attacks on real-time C37.118 data packets using a packet manipulation tool called Scapy. Using this platform, we generated C37.118 FCI and FDI datasets which are processed by multi-label machine learning classifier algorithms, such as Decision Tree (DT), k-Nearest Neighbor (kNN), and Naive Bayes (NB), to find out how effective machine learning can be at detecting such attacks. Our results show that the DT classifier had the best precision and recall rate.**

***Index Terms*—PMU, C37.118, industrial control systems, critical infrastructure security, machine learning, Scapy**

I. Introduction

Synchrophasor data measurements are critical for the power grid's operation. These measurements provide vital information about the real-time status of a power grid in order to prevent blackouts or damage to field equipment. Phasor Measurement Units (PMUs) send this telemetry data to Phasor Data Concentrators (PDCs) which are located at the substation or at the utility control center.

In the event that a synchrophasor data stream is altered, utility control centers will not be able to accurately manage the generation and load levels which could result in grid failures. For example, during the Winter Storm Uri, the Texas grid came extremely close to a complete system failure which would have damaged or destroyed essential power infrastructure [1], [2]. The winter storm caused many substations and generators to go offline, resulting in the load demand to exceed the supply of eletrical power. This forced the Electric Reliability Council of Texas (ERCOT) to implement rolling blackouts across the state for several days. Had there been an issue with ERCOT receiving accurate Synchrophasor measurement for only a matter of ten minutes, the grid would have incurred serious damage which would have left many residents without power for weeks [3].

978-0-7381-3184-9/21/33.00/$31.00 ©2021 IEEE

There have been several publications that focus on man-in-the-middle (MiTM) attacks on industrial control system (ICS) protocols [4]. They have mainly evaluated Distributed Network Protocol 3 (DNP3) [5] or Modbus [6], [7] protocols. Fewer works have tried to compromise PMU signals. Of those, most have only implemented false Global Positioning System (GPS) data that the PMU uses to synchronize its clock.

In this paper, communication packets sent from PMU units are maliciously modified while in transit. After developing software libraries for C37.118 protocol using a packet crafting library called Scapy, we inject falsified commands and data into the PMU communication channel. With these custom libraries we can change any of the fields of C37.118 packets, such as phase, voltage, timestamp, transmission off in the command frame, and the cyclic redundancy code (CRC) field. As a result, five datasets are generated where each of the fields were changed to a false value. In addition, there is a sixth dataset where both the voltage and angle fields were changed for one PMU.

Machine learning (ML) classification algorithms are very powerful tools that can be used to identify various types of attacks in a communication network [8], [9]. In this paper, we compare the precision, recall, and F-1 score of three popular supervised classification algorithms: k-Nearest Neighbor (kNN), Decision Tree (DT), and Naive Bayes (NB) [10] with the aid of Scikit-learn [11]. Each algorithm is configured to produce multi-label output [12] in the event that a packet has more than one falsified field. For instance, if an adversary injects false voltage and angle values into the same packet, the classifier should be able to label that packet as both false voltage and false angle.

These ML classifiers were chosen because, to the best of our knowledge, there are no other works that use ML classifiers to detect attacks on the C37.118 protocol. The three commonly used ML multi-label classifiers were selected to give the best chance of successfully detecting these attacks. These classifiers were also chosen for their ability to produce multi-label outputs, which is a key part of our datasets.

In summary, these are the main contributions of this work:

- To introduce a new method to perform false data injection (FDI) and false command injection (FCI) on C37.118 packets.
- To generate C37.118 datasets that can be studied by other researchers to develop or train their new detection and

Authorized licensed use limited to: Texas A M University. Downloaded on May 10,2022 at 21:43:25 UTC from IEEE Xplore. Restrictions apply.

mitigation algorithms.
- To analyze how well three common ML algorithms can classify normal C37.118 data from various types of malicious packet modifications.

The remainder of this paper is organized as follows. Section II discusses recent work on the security of C37.118. Section III reviews C37.118 packet format, explains the Scapy tool, and introduces the ML algorithms. Section IV describes our testbed, and Section V shows how Scapy is being used to inject false command and data into the communication session. The results are presented in Section VI where different ML algorithms are compared. Finally, Section VII discusses our findings and future work.

II. Related Work

Here we discuss related studies on cybersecurity for the C37.118 protocol and how they differ from our work.

The authors in [13] construct a PMU network, similar to the topology of our testbed, where a regional PDC collects the data from other substation PDCs and relays the telemetry to the control center. However, they chose to study the impact of a Denial of Service (DoS) attack on IEEE C37.118.2. They also introduce what they call Multipath-Transmission Control Protocol (TCP) based port hopping in the PMU network to mitigate the DoS attack on the C37.118.2 protocol. They verify their new Multipath-TCP scheme by simulating data packets over a wide area network (WAN).

Bhamidipati S. *et al.* [14] proposed a time authentication algorithm that thwarts external timing attacks. It is widely known that PMUs rely on high precision timestamps from GPS satellites to calibrate their internal clock. If the GPS signal is modified by an attacker, the attacker would have the opportunity to change the timestamp value. This can potentially cause over-voltage in the power line, further leading to frequency fluctuations, resulting in a cascading failure in the power grid. While our work does involve changing the timestamp, it is a bit different in that we are able to directly inject the timestamp in the PDC telemetry without the need to to alter the GPS signal.

In [15] the authors investigate Man-in-The-Middle (MiTM) and DoS against IEEE C37.118 protocol. They also provide solutions to mitigate the effects of known and unknown attacks on the protocol. While this paper is comprehensive in the multiple types of attacks that have a negative impact on the power grid, it does not implement any of them in a testbed environment. The authors have chosen to leave the verification of each of the various attacks to other researchers.

In summary, there have been studies on various ways to compromise PMU and PDC units. Some of them have even introduced detection methods for the attacks they studied. However, our attack and detection method differ from these works in two ways:

- We use an emulated network to gather and collect our C37.118 dataset, not a simulated network. This allows us to create a more realistic setup for how an attacker would compromise devices within a PDC communication network in real-time.
- Our attack not compromises the availability of packets but it also threatens confidentiality by dissecting the packets. The attack breaks the integrity of the packet by changing any of the five fields listed prior, and then updates the cyclic redundancy code (CRC) before forwarding the packet along to its target.

III. Background

This section explains the C37.118 protocol, its type of packets, the software tool that modifies the packets, and the ML algorithms used to detect the modifications.

A. IEEE C37.118 Synchrophasor Data Transfer Standard

C37.118 is an IEEE network application layer protocol for the transmission of Synchrophasor data packets from power systems [16]. This protocol is used for real-time reporting of synchronized phasor measurements from a PMU to a PDC. In a standard communication session, the PMU is the server, and the PDC is a client that subscribes to the PMU. C37.118 can be sent over Internet Protocol (IP) networks using either TCP or User Datagram Protocol (UDP) transport layer protocol, as well as over a serial connection.

There are three types of C37.118 frames: Command Frame, Configuration Frame, and Data Frame. For example, Figure 1 shows the structure of a C37.118 data frame. The configuration and command frames are very similar to the data frame with the only differences being that instead of measurement data, the command frame has command data (e.g., transmission ON or OFF) and the configuration frame has configuration data.

Within power grid networks, C37.118 packets are commonly sent as unencrypted data packets with no other forms of security. This is done to increase the reliability and reduce the complexity of the network. But this also means that any malicious user inside the network can eavesdrop as long as they has the tools to dissect or decode the C37.118 protocol.

B. Attack Tool

Scapy is a Python program written to help construct or deconstruct network packets [4]. It has many applications: network scanning, penetration testing, and false data injection. Natively, Scapy can generate and deconstruct packets for traditional Internet applications. But it allows custom packet dissectors to be written for any protocol that is based on the Open System Interconnection (OSI) model [17]. For instance, the authors in [4] developed new Scapy libraries for DNP3.

We wrote a packet dissector for the C37.118 protocol, adding a new ICS protocol to the list of protocols that Scapy can detect. It is now able to eavesdrop traffic and dissect each of the bytes in the C37.118 packet into human-readable integers, strings, flags, and timestamps. This dissector is also used to inject falsified data into the session in order to see how the PMU network will react and what values the human operator will observe. An example of the Scapy dissection of a C37.118 packet is shown in Figure 2. From top to bottom on the left side, it shows each header of the packet: Ethernet, IP, and TCP headers, with the C37.118 data frame as the payload.

Authorized licensed use limited to: Texas A M University. Downloaded on May 10,2022 at 21:43:25 UTC from IEEE Xplore. Restrictions apply.

2 Byte	2 Byte	2 Byte	4 Byte	1 Byte	3 Byte	Varies	2 Byte
Synchronization word	Framesize	PMU/DC ID number	SOC time stamp	Time quality flag	Fraction of second (raw)	Measurement data	CRC

Fig. 1. C37.118 Data Frame Structure.

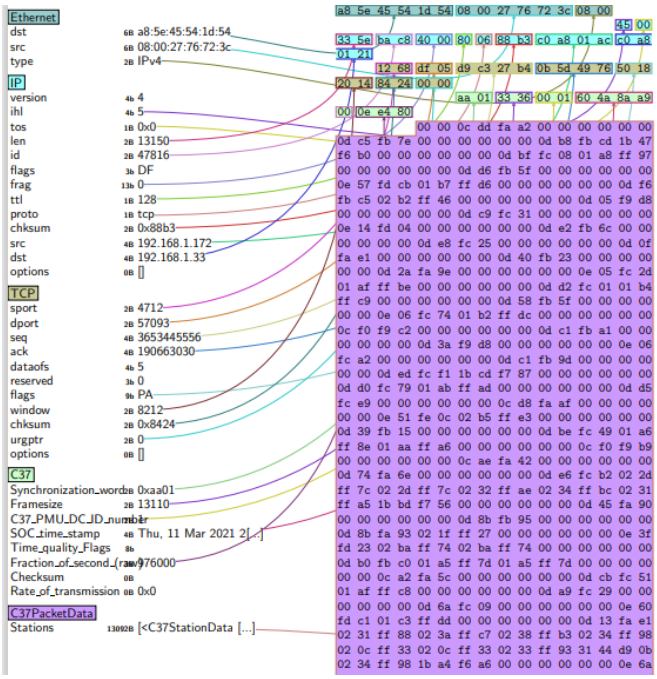

Fig. 2. Scapy Dissection of a C37.118 Packet.

The packet dissector manipulates the C37.118 hexadecimal values shown on the right side, and it supports both data and command frames.

C. Malicious Packet Detection Using Machine Learning

In this paper, three classification algorithms are used to detect different types of attacks on the C37.118 protocol:

1) k-Nearest Neighbor (kNN): It relies on the distance between new data points and training data that has already been labeled. The distance between a new data point from the nearest labeled data points determines how the new data point will be labeled. There are different distance formulations that can be used to determine the likeness of a new data point, such as Euclidean, Manhattan, and Minikowski formulas. We chose to use the Euclidean distance because it is the most commonly used and there is no clear indication from the data collected that other distance formula would increase the detection rate.

In kNN classification, the value of k is the number of nearest data points. To find the optimal value for k, multiple values have to be chosen through trial and error. The value of k that minimizes the mean square error (MSE) function can be chosen as the optimal value. This method is also known as the Elbow method.

2) Decision Tree (DT): It uses a process known as binary recursive partitioning to build a decision tree that can label data based on each value feature. There are two ways that a DT algorithm can calculate the impurity measure: Gini impurity (gini) or entropy. This impurity measure is used to determine when a branch should fracture into two nodes. In this work we tried both gini and entropy and found that the gini impurity produced a higher F1-score. For this reason, we focused on the gini impurity index.

3) Naive Bayes (NB): It relies on the Bayes algorithm to predict the label of a new data point. It uses the previous probability of an event occurring and new evidence to determine the future likelihood of an event. Bayes algorithm assumes that the features are independent of one another. However, in the real world features are almost never independent. For this reason, it is called Naive Bayes (NB).

There are many other classification algorithms that could be chosen to label the datasets we generate. We have chosen the kNN, DT, and NB algorithms because each is able to produce classify attacks with multiple labels.

IV. Experimental Setup

To study the effects of modifying the PMU telemetry, our testbed uses three virtual machines (VM), which are hosted in VirtualBox [18] on a Windows host. VirtualBox provides internal networks to connect the three VM's: the Client VM, the PowerWorld VM, and the Common Open Research Emulator (CORE) VM [19], shown in Figure 3.

The Client VM contains an open-source program called PMU Connection Tester. This tool establishes a client connection over C37.118 to either a PDC or PMU and provides a graphical user interface (GUI) where the different measurements can be observed in real-time. This VM also contains other tools such as Wireshark for capturing and exporting the C37.118 packets for further analysis.

The PowerWorld VM is used to run PowerWorld Dynamic Studio (PWDS). It simulates different power grid systems and is able to output this information as C37.118 packets. For the power grid simulation, two cases were examined: a small three-bus system with three stations and one PMU per station, and a large system with 2000 buses that represent the Texas Power Grid [20]. For simplicity, the three-bus system, shown in Figure 4, was chosen for generating the datasets presented in this paper. The reason for choosing the three-bus system over the 2000-bus system was because the data frames, which are constantly being sent by PWDS, for the three-bus system are only 100 bytes per data frame. By contrast, the data frames for the 2000-bus system are 13,164 bytes per data frame, which generated datasets that were much larger and would have taken longer to process.

Authorized licensed use limited to: Texas A M University. Downloaded on May 10,2022 at 21:43:25 UTC from IEEE Xplore. Restrictions apply.

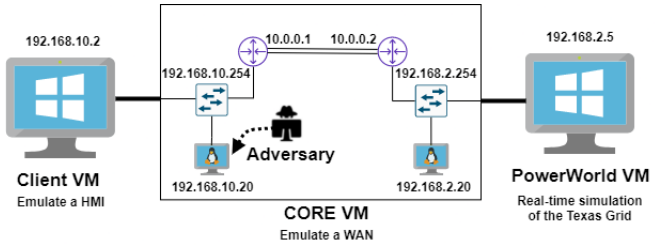

Fig. 3. Testbed Topology.

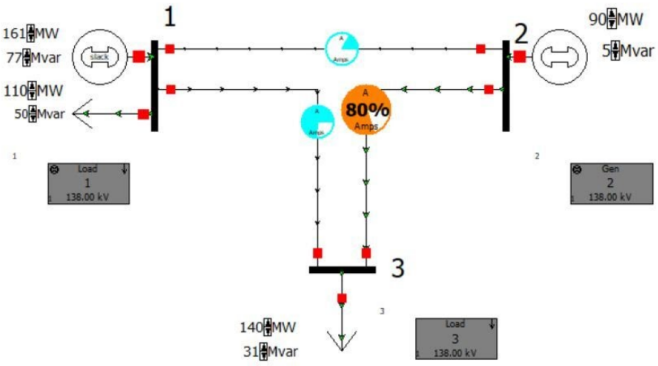

Fig. 4. Three-Bus System Simulation in the PowerWorld VM.

The CORE VM runs the CORE network emulation software for simulating the testbed network as well as running the Python scripts created to inject false commands and data into the C37.118 communication stream. Figure 3 shows the CORE VM and its emulated wide area network (WAN). The CORE network is constructed by placing and connecting different routers, switches, and hosts to mimic the desired physical network. All the devices are given IP addresses and default routes. The attack scripts run on an emulated host within the CORE network labeled *Adversary*. This allows the attacker to intercept all packets addressed to the client VM.

On the left and right sides of the CORE VM are interfaces that allow traffic from the other two VMs to flow in real-time across the emulated network. A network emulator was chosen instead of a network simulator because the emulator is more accurate in representing an actual power grid's WAN. While a network simulator would be able to generate more datasets in a shorter time, some of the steps that a real attacker would need to make in order to perform an FDI or FCI injection would be abstracted away by the network simulator. For example, datasets using simulators such as Network Simulator 3 (NS3) [21] would not have real-time delay. All physical networks and the power system have some form of latency caused by transmission delay. This delay time is needed in order for the attacker to modify the packets. Because simulators operate in steps or discrete events, moving on to the next event immediately after the previous one, there is no real-time delay between one operation and the next [22]. To make our experiment more realistic, an emulated network was necessary.

V. False Data and Command Injection

Using our new Scapy libraries [23] the route between the PowerWorld VM and the Client VM was compromised by the *Adversary* node. More specifically, the *Adversary* compromised the emulated router near the Client VM and the Client VM itself, which allows the *Adversary* to inspect all traffic that is sent to the Client VM. This type of attack is labeled an eavesdropping attack. For C37.118, the attacker can learn the names of the substations, the number of PMUs in each substation, and the rate at which the PMUs are sending telemetry to the control center. This sets up the stage for the next, active phase of the attack, which is to send false data and commands as follows:

- Phase and Voltage modification: the phase or voltage for a specific PMU is changed to zero. This would cause an operator to believe that there is a fault in the line.
- Timestamp modification: the timestamp is changed to a random value plus or minus 30 seconds from the actual time. This results in the Client HMI receiving packets with inaccurate timestamps, which can cause an operator to not be able to determine precisely when an event occurred for that PMU.
- CRC modification: the CRC value is changed to an incorrect value. This results in the client HMI discarding all C37.118 packets it receives even though the actual data is intact, and eventually the connection is disconnected. This type of attack is harder to troubleshoot since the packets are making their way from PowerWorld to the Client HMI, but the packets are appearing corrupt.
- Transmission off: any transmission ON commands sent from the client HMI are changed to transmission OFF in a C37.118 command frame. This prevents the PowerWorld server from starting a data stream. As a result, the client HMI does not receive any data packets until the attack is stopped, no matter how many times the client attempts to start the data stream.

VI. Results

We trained each ML algorithms using the same train and test split. First, each of the six datasets are compiled into one large dataset, which is referred to as the compiled dataset. This compiled dataset is randomized in order to better distribute the test and training data labels from the individual datasets. Then, the compiled dataset is split as 80% training and 20% test data. Also, the data is normalized to reduce any overfitting. Next, the three ML classifiers are trained using the training data of 50,482 C37.118 packet entries. Each ML algorithm is asked to predict the label (i.e., the type of attack or whether it is normal traffic) for the test data of 12,621 entries. Finally, the predicted labels are compared against the real labels of the test data to calculate the number of false positives, true positives, false negatives, true negatives, precision, recall, and unweighted F1-score [24].

Authorized licensed use limited to: Texas A M University. Downloaded on May 10,2022 at 21:43:25 UTC from IEEE Xplore. Restrictions apply.

A. Evaluation Metrics

There are four categories a classification can fall into: true positive, false positive, true negative, and false negative. A true positive occurs when the classification correctly labels an entry according to the test data. A false positive occurs when an entry is labeled as something it should not be. For example, if a packet is labeled as normal traffic but it is actually a compromised packet where the voltage was modified. Similarly, a true negative occurs when an entry is not mislabeled. For example, a true negative happens when a packet did not have a bad CRC and was labeled correctly as not having a bad CRC.

Precision and recall are two metrics that are important to determine the efficiency of ML algorithms. Precision is a measurement of how well the model can correctly classify positive data. Recall is a measurement for how well the model can identify true positives as opposed to false negatives. We have come to the conclusion that in this work a high recall rate is more important than a high precision rate because it is more important to detect all FCI and FDI attacks even though this may lead to a high number of false positives. The F1-score is used to make sure that there is a good balance between the precision and recall rates of a model.

B. Find the Optimal k Value

As previously discussed in Section III-C1, the kNN model requires the number of neighbors k used to classify new data points be specified prior to training the ML algorithm. The kNN model was trained multiple times with k ranging from 1 to 100. Then, the mean square error was calculated and plotted against the value of k used, as shown in Figure 5. The optimal k value was found to be $k = 7$, which means that the distance from the seven closest neighboring points will be used to classify new data points.

C. Analysis of Results

The three ML algorithms were analyzed to determine which was best at detecting the FCI and FDI attacks on C37.118 traffic. Each classification algorithm and the results for each type of attack are listed in Table I. The results in Figure 6 show that DT had the highest F1-score when used to detect false angle modification, normal traffic, transmission off, and

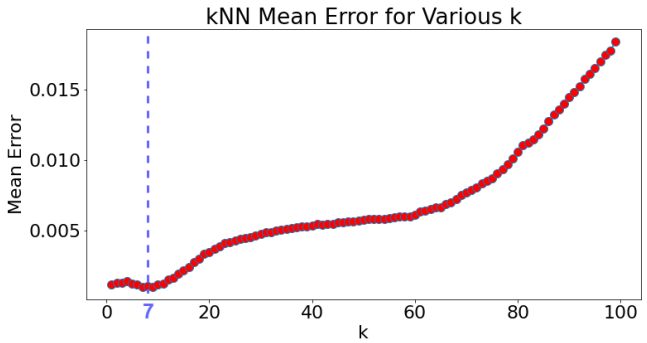

Fig. 5. Finding the Optimal Value for k Ranging from 1 to 100.

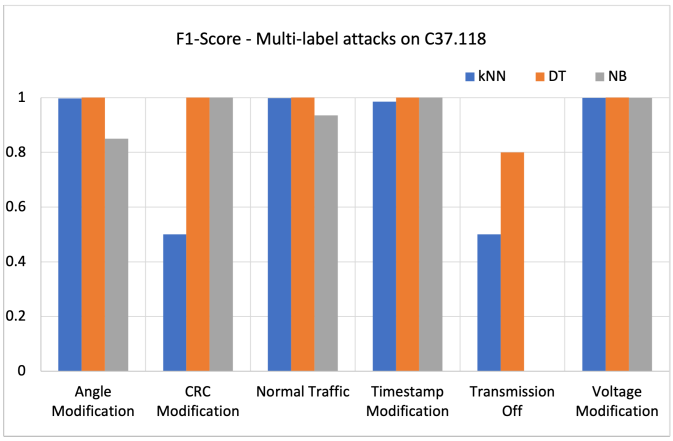

Fig. 6. F1-score Results for kNN, DT and NB Algorithms.

voltage modification. The DT algorithm tied with NB for CRC modification and timestamp modification. DT also had a recall of 100% for all six datasets, while kNN and NB only had one for a couple of attacks. Even for the hardest attack to detect, which was the Transmission Off, DT had the highest F1-score of 0.8 (80%). Transmission off was the hardest to detect because the transmission off attack prevented the data packet stream from ever being started. As a result, there are only three packets generated during this attack. Overall, as in Table I, the recall rate is fairly high for each of the algorithms, with the DT having the highest F1-score.

VII. Conclusion and Future Works

This paper presented MiTM attack scenarios on C37.118 phasor communication protocol, using custom Scapy libraries that modify the C37.118 data and command frames. It also provides datasets specific to the C37.118 protocol that can be used to test and train other multi-label ML algorithms. The results showed that of the three ML algorithms studied the DT classifier is the best at detecting FDI and FCI attacks, especially voltage, phase, CRC, and timestamp false data injection attacks in C37.118 traffic. Future works can improve on this by using more complex and advanced ML tools.

This work can also be improved by automating the dataset generation and testing to save time, prevent human error, and allow for longer testcases. This will make it easier to implement the ML algorithms in real-time and allow operators to detect attacks as they are occurring. Furthermore, this test data can be collected from the point of view of the substation PDC instead of the control center as was done in this project.

Another way this project can be improved is by randomizing the variables that are attacked. In this project, only one substation was attacked for each test; however, it will be useful to test how well the ML classifiers perform against an attack that alternates between different substations as well as the number of substations attacked.

Acknowledgment

The authors would like to acknowledge the US Department of Energy Cybersecurity for Energy Delivery Systems program

Authorized licensed use limited to: Texas A M University. Downloaded on May 10,2022 at 21:43:25 UTC from IEEE Xplore. Restrictions apply.

TABLE I
EVALUATION METRIC FOR EACH FCI AND FDI ATTACKS ORGANIZED BY TYPE OF CLASSIFIER

	FCI or FDI type	True Positive	False Positive	True Negative	False Negative	Precision	Recall	F1-Score
kNN	Angle Modification	2,131	9	10,477	4	0.9957	0.9981	0.9969
	CRC Modification	1	0	12,618	2	1	0.3333	0.5
	Normal Traffic	8,416	24	4,176	5	0.9972	0.9994	0.9983
	Timestamp Modification	963	3	11,630	25	0.9969	0.9747	0.9856
	Transmission Off	1	2	12,618	0	0.3333	1	0.5
	Voltage Modification	2,166	1	10,454	0	0.9995	1	0.9998
DT	Angle Modification	2,135	0	10,486	0	1	1	1
	CRC Modification	3	0	12,618	0	1	1	1
	Normal Traffic	8,421	0	4,200	0	1	1	1
	Timestamp Modification	988	0	11,633	0	1	1	1
	Transmission Off	2	1	12,618	0	0.6667	1	0.8
	Voltage Modification	2,166	0	10,455	0	1	1	1
NB	Angle Modification	2,135	752	9,734	0	0.7395	1	0.8502
	CRC Modification	3	0	12,618	0	1	1	1
	Normal Traffic	8,079	780	3,420	342	0.9199	0.9594	0.9351
	Timestamp Modification	988	0	11,633	0	1	1	1
	Transmission Off	0	3	12,618	0	0	0	0
	Voltage Modification	2,166	3	10.452	0	0.9986	1	0.9993

under award DE-OE0000895.

REFERENCES

[1] M. J. Patel. (2010, 10) Real-time application of synchrophasors for improving reliability. [Online]. Available: www.naspi.org/sites/default/files/reference_documents/rapir_final_20101017.pdf?fileID=519

[2] M. Perron, I. Kamwa, A. Heniche, C. Lafond, P. Cadieux, M. Racine, H. Akremi, S. Lebeau, and H.-Q. Canada, "Innovative wide-area and local voltage control of dynamic shunt compensation devices to prevent voltage collapse," 08 2016.

[3] K. Blunt and R. Gold. (2021, 2) Texas power grid was minutes from collapse during freeze, operator says. [Online]. Available: www.wsj.com/articles/texas-power-grid-was-minutes-from-collapse-during-freeze-operator-says-11614202063

[4] R. K. Rodofile, N.R. and E. Foo, "Real-time and interactive attacks on dnp3 critical infrastructure using scapy," in *13th Australasian Information Security Conference (AISC 2015)*, ser. CRPIT, I. Welch and X. Yi, Eds., vol. 161. Sydney, Australia: ACS, 2015, pp. 67–70. [Online]. Available: http://crpit.com/confpapers/CRPITV161Rodofile.pdf

[5] P. Wlazlo, A. Sahu, , Z. Mao, H. Huang, A. Goulart, K. Davis, and S. Zonouz, "Man-in-the-middle attacks and defence in a power system cyber-physical testbed," *IET Cyber-Physical Systems Theory & Applications*, June 2021. [Online]. Available: https://doi.org/10.1049/cps2.12018

[6] P. Huitsing, R. Chandia, M. Papa, and S. Shenoi, "Attack taxonomies for the modbus protocols," *International Journal of Critical Infrastructure Protection*, vol. 1, pp. 37–44, 2008.

[7] B. Chen, N. Pattanaik, A. Goulart, K. L. Butler-purry, and D. Kundur, "Implementing attacks for modbus/tcp protocol in a real-time cyber physical system test bed," in *2015 IEEE International Workshop on Communications Quality and Reliability (CQR)*, 2015, pp. 1–6.

[8] V. Gadepally, J. A. Goodwin, J. Kepner, A. Reuther, H. Reynolds, S. Samsi, J. Su, and D. Martinez, "Ai enabling technologies: A survey," *ArXiv*, vol. abs/1905.03592, 2019.

[9] P. Dixit and S. Silakari, "Deep learning algorithms for cybersecurity applications: A technological and status review," *Computer Science Review*, vol. 39, p. 100317, 2021.

[10] G. Rebala, A. Ravi, and S. Churiwala, *An Introduction to Machine Learning*. Cham Springer, 2019.

[11] (2007) Scikit-learn machine learning in python. [Online]. Available: https://scikit-learn.org/stable/index.html

[12] P. Szymański and T. Kajdanowicz, "A scikit-based Python environment for performing multi-label classification," *ArXiv e-prints*, Feb. 2017.

[13] S. M. Farooq, S. Nabirasool, S. Kiran, S. Suhail Hussain, and T. S. Ustun, "Mptcp based mitigation of denial of service (dos) attack in pmu communication networks," in *2018 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, 2018, pp. 1–5.

[14] S. Bhamidipati, K. J. Kim, H. Sun, and P. V. Orlik, "Wide-area gps time monitoring against spoofing using belief propagation," in *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2019, pp. 1–8.

[15] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, B. Pranggono, P. Brogan, and H. F. Wang, "Intrusion detection system for network security in synchrophasor systems," in *IET International Conference on Information and Communications Technologies (IETICT 2013)*, 2013, pp. 246–252.

[16] "Ieee standard for synchrophasors for power systems," *IEEE Std C37.118-2005 (Revision of IEEE Std 1344-1995)*, pp. 1–65, 2006.

[17] L. L. Perterson and B. S. Davie, *Computer Networks: A Systems Aproach*. Elsevier Science, 2011, vol. 5th Edition.

[18] (2021) Virtualbox. [Online]. Available: www.virtualbox.org

[19] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "Core: A real-time network emulator," in *MILCOM 2008 - 2008 IEEE Military Communications Conference*, 2008, pp. 1–7.

[20] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3258–3265, 2017.

[21] F. Zhang, H. A. D. E. Kodituwakku, J. W. Hines, and J. Coble, "Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4362–4369, 2019.

[22] I. McGregor, "The relationship between simulation and emulation," in *Proceedings of the Winter Simulation Conference*, vol. 2, 2002, pp. 1683–1688 vol.2.

[23] R. R. S, R. R, M. Moharir, and S. G, "Scapy- a powerful interactive packet manipulation program," in *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, 2018, pp. 1–5.

[24] K. P. Shung. (2020, 4) Accuracy, precision, recall or f1? [Online]. Available: towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Authorized licensed use limited to: Texas A M University. Downloaded on May 10,2022 at 21:43:25 UTC from IEEE Xplore. Restrictions apply.