Evaluating Synthetic Datasets for Training Machine Learning Models to Detect Malicious Commands

Jia Wei Teo*
National University of Singapore
Singapore
e0415724@u.nus.edu

Sean Gunawan*
Singapore University of Technology and Design
Singapore
sean_gunawan@mymail.sutd.edu.sg

Partha P. Biswas, Daisuke Mashima
Illinois at Singapore Pte Ltd
Singapore
{partha.b, daisuke.m}@adsc-create.edu.sg

***Abstract*—Electrical substations in power grid act as the critical interface points for the transmission and distribution networks. Over the years, digital technology has been integrated into the substations for remote control and automation. As a result, substations are more prone to cyber attacks and exposed to digital vulnerabilities. One of the notable cyber attack vectors is the malicious command injection, which can lead to shutting down of substations and subsequently power outages as demonstrated in Ukraine Power Plant Attack in 2015. Prevailing measures based on cyber rules (e.g., firewalls and intrusion detection systems) are often inadequate to detect advanced and stealthy attacks that use legitimate-looking measurements or control messages to cause physical damage. Additionally, defenses that use physics-based approaches (e.g., power flow simulation, state estimation, etc.) to detect malicious commands suffer from high latency. Machine learning serves as a potential solution in detecting command injection attacks with high accuracy and low latency. However, sufficient datasets are not readily available to train and evaluate the machine learning models. In this paper, focusing on this particular challenge, we discuss various approaches for the generation of synthetic data that can be used to train the machine learning models. Further, we evaluate the models trained with the synthetic data against attack datasets that simulates malicious commands injections with different levels of sophistication. Our findings show that synthetic data generated with some level of power grid domain knowledge helps train robust machine learning models against different types of attacks.**

I. Introduction

Smart grids are equipped with information and communication technology for remote monitoring and control of power grid devices. The smart grid consists of two layers, cyber and physical systems [1]. In the cyber layer, a vast amount of intelligent devices form a cyber network to monitor, control and protect the physical systems. By switching from traditional electric grid to smart grid, there are benefits to be reaped including but not limited to more efficient power transmission, reduced management costs and consumer prices, and improved integration with other power systems such as renewable energy systems [1]. This innovation is becoming increasingly popular among companies and governments worldwide and locally in Singapore, with Jurong Town Corporation developing one such grid for the Punggol Digital District [2].

The substation is a critical entity in the smart grid and it primarily consists of transformers, circuit breakers, and switch gears. Substations enable the transformation of electricity from generators through different voltage levels for efficient delivery to the end consumers. [3]. Intelligent electronic devices (IED) are installed in substations, and these digital devices form a communication network to facilitate remote control, to automate fault responses, and to optimize power grid operations. The IEDs record measurements and the state of the substations, and enable the remote control of substations through remote commands such as those for switching on/off of circuit breakers issued by a control center.

With such digitalization, there is an increased risk of cyber attacks that could subvert power grid services. Any malicious intrusion in the communication channel may jeopardise a part or whole of the power gird, for instance by means of injection of malicious control commands. Such an attack can be initiated in various ways such as through the network or from the control center. The integrity of the network between the control center and substations can be compromised through man-in-the-middle (MITM) attack or replay attacks if there are insufficient security implementations [4]. An attack initiated through the control center can be achieved through many ways, from having a malicious employee in the control center performing hostile actions to the installation of rootkits to gain access to the system through privilege escalation. One such real-world cyber attack experienced is in the Ukraine Power Plant Attack in 2015, where the attackers exploited the remote control interface and manipulated a large number of circuit breakers to be opened. As the result, around 30 substations went offline for hours, causing a massive blackout [5].

There are existing defense and mitigation techniques to prevent command injection attacks, mainly using physics-based or rule-based approaches. Physics-based methods calculate, based on the power system physical laws, the estimated values of the new state to see if the incoming command were to be executed, and then determine if the system violates any stability constraint (e.g. power flow limit in transmission lines, low/high bus voltage, etc.). This can be achieved through on-the-fly power flow simulation [6]. However, such physics-based methods often suffer from high latency [6] and may perform poorly with incomplete data (e.g., missing measurements, which often happen in a real-world operation). On

* Authors were with Illinois at Singapore Pte Ltd when the work was done.

978-1-6654-3254-2/22/$31.00 ©2022 IEEE

the other hand, rule-based approaches consist of firewalls and intrusion detection systems (IDS). They work on a specific set of cyber rules to allow legitimate traffic inside a network. By monitoring and analyzing the network traffic in real-time, IDS and firewalls identify and block an event that does not satisfy the security policy of the system. However, the major drawback of the rule-based techniques is that they may fail to counter attacks that follow the normal communication model, thus unable to counter cases like the Ukraine incident where a legitimate control center machine sends out malicious commands. Message authentication and cryptographic protection (e.g., [7], [8]) would also fail if an authorised user mistakenly (or perhaps intentionally) issues a harmful command to the system or if a legitimate device is compromised.

With the limitations of physics-based and rule-based approaches mentioned above, the use of machine learning (ML) provides a potential solution due to its promising computational and reasoning capabilities. However, in order to implement the ML models effectively, we need substantial datasets to train and test the models. As there is limited coverage of ML in the field of command authentication, this leads to insufficient real-world command injection attack data to train the ML models. Furthermore, since power grid operators are often not willing to disclose data, even normal data is not made available to the research community. In this paper we explore the use of power flow simulator to synthetically generate training and test datasets for the ML models. We consider different ways of generating such datasets, with different levels of domain knowledge utilized, and then train several ML models. We then create multiple datasets of simulated attacks to evaluate the effectiveness of each training dataset. Through experiments, we also demonstrate the low latency for attack detection. It is important to note that the main emphasis of this paper is on the generation and evaluation of synthetic datasets for ML models' training and testing and not on the development or fine-tuning of ML models. We summarize the following contributions of our work:

- First, we discuss the use of open-source power flow simulator (namely Pandapower [9]) to create synthetic datasets, which are customized with different levels of power system domain knowledge, for a 3-substation model for the command authentication case study.
- Second, we generate synthetic attack datasets that assumes different level of sophistication of attackers, using a power flow simulator.
- Third, we train various supervised machine learning models on the generated training datasets for detection of malicious commands. The performance and latency of the models are evaluated against the attack datasets.

This paper is organized as follows. In Section II, we discuss the related work. Section III provides an overview of the system and threat models we assume. Section IV presents the test network and the details of synthetic training/attack dataset generation. Section V discusses the different ML models implemented and the test datasets used to evaluate the ML models. Section VI evaluates the effectiveness of the ML models. Finally, we conclude the paper with future research directions in Section VII.

II. Related Work

With no proper command authentication, an adversary can effectively violate the availability of the system by opening circuit breakers and remotely switching substations off, resulting in power outage as shown in the infamous Ukraine power grid attack in 2015 [5]. Additionally, the aurora vulnerability showed that cyberattack through malicious commands can destroy physical components of the electric grid [10]. The research work on the defense against malicious command injection attacks tries to differentiate between legitimate and illegitimate control commands. A cyber-based IDS monitors and analyzes the network traffic in real time, identifying/blocking events that does not satisfy the security policy. The cyber-based IDSes such as signature-based, are common in detecting attacks in various cyber-physical systems including smart grids [11]. However, such cyber-based approaches do not take account of up-to-date power grid state information for context-aware attack detection.

Physics-based methods for analysis and security of power grid have also been widely adopted in the past. Mashima *et al.* [12] provided a framework called active command mediation defense (ACMD). ACMD is deployed in substations for securing remote control interface. This mechanism offer an addition layer of defense against attacks that bypass other cyber security measures. Meliopoulos *et al.* [13] implemented a physics-based approach for command authentication using distributed dynamic state estimation that enabled faster than real time simulation. Zeng *et al.* [14] proposed a physics-constrained vulnerability assessment methodological framework to detect stealthy false data injection (FDI) attack. They also proposed a physics-constrained robustness verification [15] that evaluates the vulnerability of intelligent stability assessment (ISA) for power systems and provide suggestions to select the ML models. Remotely issued control commands were authenticated by incorporating a delay [16] and using the latency to simulate the power system dynamics [17]. Use of power system dynamics simulation for malicious command detection is studied in [6]. While the accuracy is promising, the simulation takes long time (e.g., around 1 second for relatively small systems), which has motivated our work.

The application of ML in IDS are done in several contexts, e.g., in IoT based systems [18], for SCADA systems [19], and even for smart cities and related infrastructure [20]. However, most of these works focused on False Data Injection (FDI) and not malicious command injection. Esmalifalak *et al.* [21] attempted to detect stealthy FDI attacks using a support vector machine (SVM) based technique and a statistical anomaly detection approach. They showed that SVM outperformed statistical approach when the model was trained with sufficient data. He *et al.* [22] proposed a conditional deep belief network (CDBN) based detection scheme that extracted temporal features from distributed sensor measures. The proposed detection scheme is robust against various attack measurements and environmental noise. Karimipour *et al.* [23] proposed a computationally efficient and independent mechanism using feature extraction scheme and time series partitioning to detect FDI attacks.

To implement ML models, we need to have substantial datasets to train and test our models. These datasets can be derived from real-world datasets or created synthetically. The data can be synthetically generated using rule-based or ML-based approaches. Paul *et al.* [24] used load ranking and K-means clustering algorithms as two different approaches to attack smart grid for selecting the most vulnerable transmission lines to create contingencies. Ni *et al.* [25] proposed another reinforcement learning based sequential line switching attack to initiate blackout.

In general, the ML-based approach can provide more efficient detection compared to rule-based/physics-based approach. The requirements of accurate and fast detection of cyberattacks in electrical substations motivated us to implement robust ML models that are trained/tested using synthetic datasets which have been created emulating real-world threats.

III. Threat and System Models

Whilst smart grid systems can be protected in multiple layers, we focus on the mitigation of cyber attacks in modernized substations. More specifically, this paper focuses on malicious command injection attacks that abuse the remote control interface in substations. The consequence of such an attack is experienced in the Ukraine Power Plant Attack [5], [26]. Engineering PC is another potential entry point for the attacker. The engineering PC is connected to industrial control system devices like IEDs and PLCs, on which the attacker can install a malware, similar to what has been done with the Stuxnet attack [27]. IEDs supply chain vulnerability can also be exploited to install malware. Once gained foothold inside the substation, the attacker can use the compromised device(s) to inject malicious commands or measurement data. Malicious commands could also come from the compromised control center (or man-in-the-middle attack).

In the context of this work, we assume an intrusion detection system (IDS) deployed in a field substation to monitor network traffic and detect suspicious activities, such as malicious command injection. Such an IDS can be connected to a mirror port of switch for passively (i.e., in a non-intrusive manner) monitoring network traffic conveying power grid measurements as well as control commands (e.g., open/close of circuit breakers for protection) sent by a SCADA HMI, protection relays, and PLCs.

Inside the intrusion detection system, physics-based detection, e.g., one proposed in [6], is implemented. The ML-based attack detection module can be additionally deployed in front of it to "pre-filter" the incoming command packets to minimize the invocation of the computationally heavy physics-based detection so that overall latency can be reduced. The conceptual module architecture of the intrusion detection system is shown in Fig 1. We assume that the IDS can passively collects both command packets as well as measurement packets transmitted in the substation network. The measurements at each time slot are collectively used as a feature vector for the both of machine learning based and physics/rule-based detection modules, along with the incoming control command to be evaluated.

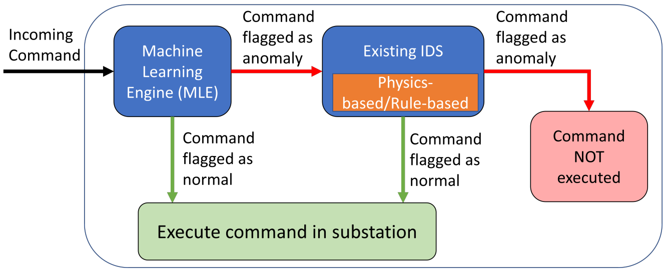
Incoming Command
Machine Learning Engine (MLE)
Command flagged as anomaly
Existing IDS
Physics-based/Rule-based
Command flagged as anomaly
Command flagged as normal
Command flagged as normal
Command NOT executed
Execute command in substation

Fig. 1. Conceptual IDS model using ML as first layer of defense.

IV. Synthetic Training Datasets

A. 3-substation Test Network for Care Study

We consider a 3-substation network model for our study purpose, and the same is shown in Fig. 2. The model is typical of an industrial customer where multiple substations are interconnected with redundant feeders. The grid feeds power to substation S/S-1 which connects substations S/S-2 and S/S-3 with redundant feeders (lines). Each substation has two transformers to step down the voltage to the utility level. Every single feeder is equipped with at least one IED for controlling and monitoring tasks. There is a total of 32 IEDs in all these substations. The IED interconnection and more details related to the single line diagram of single substation can be found in [28]. As the substations are located nearby, a single intrusion detection device would be able to monitor the network traffic and flag for any anomaly.

B. Dataset Features

In this network there are 34 switches or circuit breakers (denoted by *CB-x*), 6 transformers (denoted by *Tx*), 18 active and reactive loads (denoted by Lines *Lx*) and 2 grid connections (denoted by *Feeder x*). There are a total of 78 features (i.e., $34 + 6 + 18 \times 2 + 2$). A violation of certain conditions in the test network will be considered as an anomalous datapoint in our dataset. There are different types of violations, viz., invalid grid configuration, lines/transformer overload, bus over/under voltage, and number of open switches. All the datasets are generated using Pandapower simulator [9].

C. Generating Training Datasets

Training datasets consist of a set of power flow measurements of the power grid model of interest, as well as device status (namely circuit breaker open/close). In other words, we assume that the intrusion detection system collects up-to-date power grid measurements and circuit breaker status by overhearing the SCADA communication. When the IDS observes a

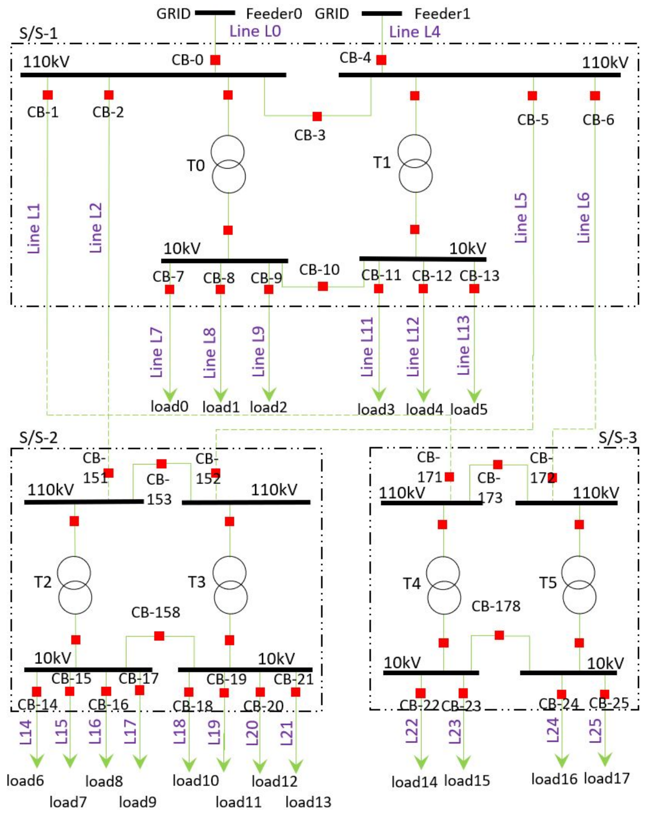

Fig. 2. One-line diagram of the test network with 3 substations.

packet carrying a control command, the corresponding device status (only circuit breaker status in the scope of this paper) is overwritten accordingly to evaluate the command is reasonable given the current power grid status.

We utilize a power flow simulator, namely Pandapower for this study, to generate training datasets. The input power system topology (i.e., feeder and circuit breaker status) and output of power flow simulation together serve as a datapoint. At the high-level, we generate a large-enough set of different power system topology and load profile settings and then run power flow simulation for each. Then, if we observe any violation of stability conditions (e.g., over/under-voltage and over/under-current) in the simulation result, the datapoint is labeled "positive". When preparing the input system typologies, we incorporate different levels of domain knowledge.

One primitive way to generate the set of input configurations (a set of circuit breaker status) is to randomly pick open/close status for each circuit breaker in the model. Besides, we add variations in the load profile. While this is feasible to generate large datapoints, this training data didn't perform well based on our preliminary study. Nonetheless, this dataset (called *Random* dataset) is used as a basis of comparison for the performance of other training datasets. In the rest of this section, we discuss multiple practical strategies to construct training datasets.

1) Minimally-constrained Random (MR) Dataset: In *Minimally-constrained Random* dataset, we enforce that at least 1 incoming feeder/transformer is online and always connected in each substation. This is based on assumption that a situation where all of these are disconnected is highly unusual and should be immediately flagged anomalous. Hence, the datapoints in this dataset adhere to valid grid configurations and do not violate this constraint. Circuit breakers connected to the loads are then toggled on/off randomly. This may result in the some of the datapoints violating the stability conditions (e.g., over/under-voltage and over/under-current) in the simulation result, leading to a "positive" result. In short, this dataset utilizes minimal domain knowledge to narrow down the space for randomly generating power grid configurations used as input.

2) Incrementally-tweaked (IT) Dataset: As the substations are in a continuous and dynamic environment, the switches/loads and overall state of the substation may change over time. To incorporate such changes, we first generate a valid starting point (i.e., without any violation) for any grid configuration before applying "tweaks" to modify the the power grid topology. Tweaks are possible ways to make the grid incrementally approach states that violate power grid stability conditions, e.g., by connecting a load, disconnecting a feeder, and disconnecting a generator. We apply one tweak at a time, and the resultant modification of the grid constitutes a new datapoint. The process is repeated till we reach a violation or we run out of options to tweak. Hence, the *Incrementally-tweaked* dataset in the end consists of sets of datapoints, where each set will start with a valid starting datapoint followed by the datapoints with tweaks sequentially applied.

3) Enumerated Normal (EN) Dataset: The *Enumerated Normal* dataset is an improved variant of the *Incrementally-tweaked* dataset. It provides a wider coverage of different grid permutations as well as load profiles. Instead of having any random valid configuration as a starting point for each set, we enumerate all possible valid configurations. This step requires more domain knowledge than than *Incrementally-tweaked* dataset, and in the context of the 3-substation model used for our case study, it can be done as follows. In each substation, among incoming lines and bus coupler switches, we need at least 2 of these switches to be closed for any valid configuration. For example, to ensure a valid grid configuration, we need CB-3 and either CB-0 or CB-4 to be closed. We get a total of 4,096 valid configurations ((4 possible permutations of 2 out of 3 switches closed) ^ (6 combinations of incoming lines/transformers and bus coupler)). For each 4,096 valid configurations, we generate multiple different load profiles for the configuration as our starting point, before applying the "tweaks" sequentially (same as *Incrementally-tweaked (IT)* dataset) to generate the remaining datapoints.

4) Enumerated Normal With Random Toggle (ENRT) Dataset: Lastly, the *Enumerated Normal with Random Toggle (ENRT)* dataset has a slight modification to the *Enumerated Normal* dataset. *ENRT* adds an additional step after the generation of *Enumerated Normal* dataset. After applying a tweak, we randomly toggle any circuit breaker status. This ensures that the dataset contains some invalid configuration for the machine learning models to learn.

V. Generating Test/Attack Datasets

Test datasets primarily consist of datapoints that violate the power grid conditions and each datapoint provides a realistic state of the network that has potentially been modified by a malicious command injection. We explore the *Unconstrained*, *Special*, and *Attack* datasets.

1) Unconstrained: The motivation for the *Unconstrained* dataset is simple: randomly switching on and off any switches, transformers and feeders to simulate a grid configuration for test data. This simulates the behaviour of an attacker maliciously disrupting the grid by toggling on/off random switches to cause disruption.

2) Sequential: The *Sequential* dataset is generated in a way similar to *MR* dataset (see Section IV) in that there is always a connection between at least 1 feeder and all non-load buses. The datapoints are generated as sets, each of which is consisting of 1 valid starting datapoint where the substations are operating safely, followed by anomalous datapoints where they violate power flow constraints. These anomalous datapoints are created by applying tweaks similar to *IT* dataset. This dataset simulates a smarter attacker causing a violation in the grid by sequentially toggling on/off circuit breakers, transformers or feeder lines one by one.

3) Strategic: The dataset simulates the idea of an attacker maliciously opening switches such that it causes invalid grid configuration state. Here we assumes that the attacker knows the system topology and strategically attacks the circuit breakers to cut off power supply to certain sections of the network. In the test network, this occurs when there are more than 2 switches being opened out of the 3 switches of incoming lines/transformers and bus coupler. For example, an invalid configuration and violation occurs when CB-10 and T0 are opened simultaneously. This is because there will be disconnection and supply disruption to lines L7-L9. The *Strategic* dataset generates all possible grid configurations. At each combination of incoming lines/transformers and bus coupler, we have 8 possible permutations of the 3 switches being closed and opened (2^3 permutations). As we have 6 sets of these combinations, we will create $8^6 = 262,144$ possible unique grid configurations. Out of 8^6 datapoints, there will be a total of $8^6 - 4,096$ (valid grid configurations), i.e., 258,048 invalid grid configuration datapoints.

VI. Evaluation and Preliminary Results

A. Accuracy of Models

We evaluate several supervised ML models, namely Logistic Regression, Decision Tree, Random Forest, Adaptive Boosting (AdaBoost), Extreme Gradient Boosting (XGBoost), and Artificial Neural Networks (ANN), using open source libraries such as scikit-learn [29] and TensorFlow [30]. Since our goal is not on finding optimal ML model, the coverage is not exhaustive. The ML models are selected from a diverse range of complexities, with Logistic Regression and Decision Trees chosen as models with low complexities since these models are simple and easy to understand. Adaboost, Random Forest and XGBoost are different ensemble methods whereby multiple models are created and combined to produce improved results. The three ML models differs in their ensemble techniques and are models of mid complexities. Lastly, Artificial Neutral Network (ANN) is a model of high complexity and it simulates the behavior of biological systems composed of "neurons", comprising of an input node layer, one or more hidden node layers, and an output node layer. Each ML model is trained with one of the training datasets discussed in Section IV, and tested on 3 test datasets discussed in Section V. As such, we will have a total of 4 training datasets × 6 ML models × 3 test datasets, i.e., 72 sets of results. However, due to space constraint, we will discuss only the prominent results achieved by certain models and datasets.

We test the performance of some selected models on the generated datasets to evaluate the detection accuracy of malicious command injection attacks. When evaluating the results, we put more emphasis on false negatives, i.e., whether the malicious command slips through the ML algorithm detection scheme. If a malicious command is treated as a normal datapoint by the ML model, it will be accepted as a valid command, leading to negative consequences to the power grid. On the other hand, having false positives will only lead to a higher latency as the command will be fed into the existing IDS to be checked again. Hence, false negatives are costlier than false positives.

TABLE I
Results of ML models trained with *ENRT* dataset and tested on *Sequential* dataset.

Model	**Accuracy**	**Precision**	**Recall**	**$F1$ score**
Logistic Regression	90.36%	0.9938	0.8768	0.9316
Decision Tree	75.90%	0.9974	0.6803	0.8089
Random Forest	94.14%	0.9986	0.9230	0.9593
AdaBoost	45.66%	1	0.2755	0.4320
XGBoost	97.16%	0.9986	0.9634	0.9807

TABLE II
Results of ML models trained with *ENRT* dataset and tested on *Strategic* dataset.

Model	**Accuracy**	**Precision**	**Recall**	**$F1$ score**
Logistic Regression	90.37%	0.9953	0.9064	0.9488
Decision Tree	96.41%	1	0.9635	0.9814
Random Forest	98.37%	1	0.9835	0.9916
AdaBoost	94.25%	1	0.9415	0.9699
XGBoost	99.30%	1	0.9929	0.9964

In terms of model performance seen in Table I and Table II, XGBoost shows the most promising results across training with 4 different training datasets, with the highest accuracy and recall consistently. Random Forest also performs consistently well as compared to the other models. Accuracy of Decision Tree and AdaBoost varies significantly depending on the test data. This may be because *ENRT* training dataset and *Strategic* attack dataset were created based on the similar idea, thus resulting in better accuracy.

TABLE III
MICRO AVERAGE OF RANDOM FOREST MODELS TRAINED WITH DIFFERENT DATASETS AND TESTED ON AGGREGATED ATTACK DATA

Training Dataset	Accuracy	Precision	Recall	$F1$ score
Random	74.54%	0.9924	0.7263	0.8387
MR	96.46%	0.9858	0.9774	0.9816
IT	96.44%	0.9858	0.9773	0.9815
EN	86.40%	0.9999	0.8694	0.9301
ENRT	98.20%	0.9999	0.9815	0.9906

TABLE IV
MICRO AVERAGE OF XGBOOST MODELS TRAINED USING DIFFERENT TRAINING DATASETS AND TESTED ON AGGREGATED DATA

Training Dataset	Accuracy	Precision	Recall	$F1$ score
Random	74.53%	0.9920	0.7264	0.8387
MR	98.16%	0.9860	0.9951	0.9906
IT	97.48%	0.9860	0.9879	0.9870
EN	93.18%	0.9999	0.9297	0.9635
ENRT	99.21%	0.9999	0.9919	0.9959

To evaluate accuracy against mixture of attack patterns, using the aggregated results of all 3 testing datasets, we show the micro average for Random Forest (Table III) and XGBoost (Table IV). We can see that *ENRT* training dataset performed the best overall in both cases, followed by *MR* training dataset.

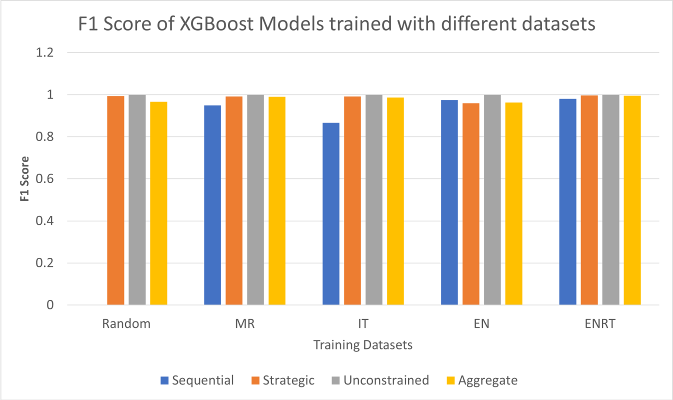

Fig. 3. Comparison of F1 Score

The F1 score is defined as the harmonic mean of precision and recall, with the highest possible value being 1, indicating perfect precision and recall. The comparison of F1 score among different combination of training and testing datasets are shown in Figure 3. Here, we also included a *Random* training dataset that is created by randomly selecting circuit breaker status without any restriction to act as a benchmark. When *Random* training dataset is used, the F1 score when tested against *Sequential* attack data is 0, implying that it failed to detect any positive samples. On the other hand, it showed comparable accuracy against *Strategic* and *Unconstrained* attack data. This affected the overall accuracy in the case of aggregated attack data not only with XGBoost but also Random Forest (Table III, Table IV). We think it is because *Sequential* attack data in nature includes positive examples with minimal deviation from normal (negative) samples. This implies that adding some constraints backed by power system domain knowledge can generate more robust ML models.

TABLE V
RESULTS OF ANN MODEL TRAINED USING *ENRT* DATASET AND TESTED ON DIFFERENT TEST DATASETS.

Test Dataset	Accuracy	Precision	Recall	$F1$ score
Unconstrained	99.98%	0.9998	1	0.9999
Strategic	97.99%	1	0.9796	0.9897
Sequential	99.00%	0.9908	0.9699	0.9802

Using TensorFlow, we experimented with ANN. Grid search is performed to achieve the optimal hyperparameters (i.e., number of layers, number of nodes per layer, loss function, activation function, epochs). Cross-validation is performed to gauge the performance of the model on unseen data and ensure that the model does not overfit. Fig. 4 shows the loss, precision, and recall curves for training and validation. All three graphs indicate a good fit as the validation curves follow all the training curves closely with minimal gap and reach a point of stability. The ANN model shows promising results (Table V) for all attack datasets.

B. Latency for Malicious Command Detection

Aside from the accuracy of the ML models, the latency of prediction is evaluated as well, because lower latency is one of the motivation for using ML-based malicious command detection over physics-based approaches. The machine used for testing of the ML models is equipped with Intel Core i7-8586U (8 Cores), and 16GB RAM. XGBoost is the fastest ML model, taking only **0.763 microseconds** to evaluate a single datapoint, while ANN takes around **17.5 microseconds**. Both models incurs significantly lower latency than the physics-based malicious command detection scheme (e.g., [6], which takes around **1s**), making ML-based approaches more suitable for online, real-time processing.

VII. CONCLUSIONS

In this paper, we conducted a preliminary study on feasibility and effectiveness of synthetic data generation for using machine learning technologies for malicious command detection in smart grid. Specifically we used multiple strategies to generate synthetic training datasets using an open-source power flow simulator with different levels of domain knowledge incorporated. We further developed multiple attack datasets for testing, each of which assumes different sophistication and strategy of attackers. Based on the simulation experiments, we showed that infusing domain knowledge into training data generation helps us generate more robust ML models against different kinds of attacks and that ML-based attack detection can be done significantly faster, without compromising accuracy, than traditional, physics-based approaches.

A major component of our future work is extensive evaluation. Following the same framework, we intend to develop more test data to increase the test coverage by implementing more diverse environment and constraints. One possible way is through the means of human attackers, for instance by having Capture-the-Flag (CTF) events.

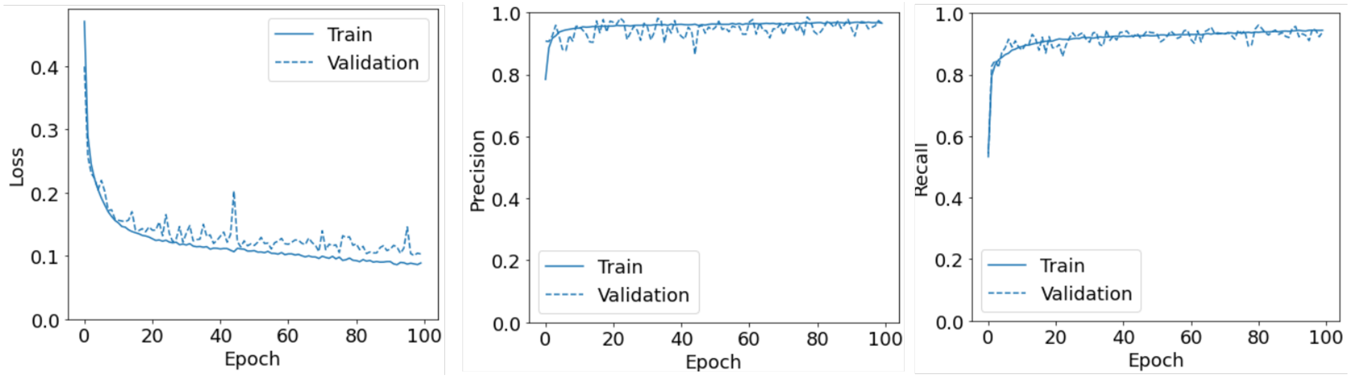

Fig. 4. ANN training validation loss, precision and recall curves

Acknowledgment

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

[1] U.S. Department of Energy. (n.d.). Smart Grid: The Smart Grid — SmartGrid.gov. SmartGrid.Gov. https://www.smartgrid.gov/the_smart_grid/smart_grid.html

[2] JTC Corporation. (2018, October 31). JTC — JTC and SP Group to Develop and Operate Singapore's First Smart Grid for Business Parks at Punggol Digital District. JTC. https://www.jtc.gov.sg/news-and-publications/press-releases/Pages/20181031(PR1)-.aspx

[3] Energy Education. (2020, April 28). Electric Substation https://energyeducation.ca/encyclopedia/Electrical_substation

[4] B. Kang, P. Maynard, K. McLaughlin, S. Sezer, F. Andren, C. Seitl, F. Kupzog, and T. Strasser. Investigating cyber-physical attacks against IEC 61850 photovoltaic inverter installations. In Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on, pages 1–8. IEEE, 2015.

[5] K. Zetter. (2016, March 3). Inside the cunning, unprecedented hack of Ukraine's power grid. Wired. Retrieved April 2, 2022, from https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/

[6] Mashima D, Chen B, Zhou T, Rajendran R, Sikdar B. Securing substations through command authentication using on-the-fly simulation of power system dynamics. In2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) 2018 Oct 29 (pp. 1-7). IEEE.

[7] Esiner, E., Tefek, U., Erol, H.S., Mashima, D., Chen, B., Hu, Y.C., Kalbarczyk, Z. and Nicol, D.M., 2022. LoMoS: Less-online/More-offline Signatures for Extremely Time-critical Systems. IEEE Transactions on Smart Grid.

[8] Tefek, Utku, Ertem Esiner, Daisuke Mashima, Binbin Chen, and Yih-Chun Hu. "Caching-based Multicast Message Authentication in Time-critical Industrial Control Systems." In IEEE INFOCOM 2022-IEEE Conference on Computer Communications, pp. 1039-1048. IEEE, 2022.

[9] Pandapower pandapower. http://www.pandapower.org/

[10] INCIBE. (2020, March 4). Aurora Vulnerability: Origin, explanation and solutions. INCIBE. Retrieved April 2, 2022, from https://www.incibe-cert.es/en/blog/aurora-vulnerability-origin-explanation-and-solutions

[11] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," International Journal of Electrical Power & Energy Systems, vol. 99, pp. 45–56, 2018.

[12] Gunathilaka P, Mashima D, Chen B. Softgrid: A software-based smart grid testbed for evaluating substation cybersecurity solutions. InProceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy 2016 Oct 28 (pp. 113-124).

[13] S. Meliopoulos, G. Cokkinides, R. Fan, L. Sun, and B. Cui, "Command authentication via faster than real time simulation," 2016 IEEE Power and Energy Society General Meeting (PESGM), 2016, pp. 1-5, doi: 10.1109/PESGM.2016.7741974.

[14] L. Zeng, M. Sun, X. Wan, Z. Zhang, R. Deng and Y. Xu, "Physics-Constrained Vulnerability Assessment of Deep Reinforcement Learning-based SCOPF," in IEEE Transactions on Power Systems, 2022, doi: 10.1109/TPWRS.2022.3192558.

[15] Z. Zhang, M. Sun, R. Deng, C. Kang and M. -Y. Chow, "Physics-Constrained Robustness Verification of Intelligent Security Assessment for Power Systems," in IEEE Transactions on Power Systems, doi: 10.1109/TPWRS.2022.3169139.

[16] T. T. Kim, and H. V. Poor, "Strategic protection against data injection attacks on power grids," IEEE Transactions on Smart Grid, vol. 2, no. 2, pp. 326–333, 2011

[17] E. Drayer, and T. Routtenberg. (2019, August 23). Detection of false data injection attacks in Smart Grids based on Graph Signal Processing. Retrieved April 2, 2022, from https://arxiv.org/pdf/1810.04894.pdf

[18] M. N. Napiah, M. Y. I. B. Idris, R. Ramli, and I. Ahmedy, "Compression header analyzer intrusion detection system (cha-ids) for 6lowpan communication protocol," IEEE Access, vol. 6, pp. 16 623–16 638, 2018.

[19] W. Ren, T. Yardley, and K. Nahrstedt, "Edmand: Edge-based multi-level anomaly detection for SCADA networks," in 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids. IEEE, 2018, pp. 1–7.

[20] V. Garcia-Font, C. Garrigues, and H. Rifa-Pous, "A comparative study of ‘ anomaly detection techniques for smart city wireless sensor networks," sensors, vol. 16, no. 6, p. 868, 2016.

[21] M. Esmalifalak, L. Liu, N. Nguyen, R. Zheng, and Z. Han. Detecting stealthy false data injection using machine learning in smart grid. IEEE Systems Journal, 11(3):1644–1652, 2014.

[22] Y. He, G. J. Mendis, and J. Wei. Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism. IEEE Transactions on Smart Grid, 8(5):2505–2516, 2017.

[23] H. Karimipour, A. Dehghantanha, R. M. Parizi, K. K. R. Choo, and H. Leung. A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids. IEEE Access, 7:80778–80788, 2019

[24] S. Paul, M. R. Haq, A. Das, and Z. Ni. A comparative study of smart grid security based on unsupervised learning and load ranking. In 2019 IEEE International Conference on Electro Information Technology (EIT), pages 310–315. IEEE, 2019.

[25] Z. Ni, S. Paul, X. Zhong, and Q. Wei. A reinforcement learning approach for sequential decision-making process of attacks in smart grid. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1–8. IEEE, 2017.

[26] Defence Use Case, "Analysis of the cyber attack on the Ukrainian power grid," 2016.

[27] J. P. Farwell, and R. Rohozinski, "Stuxnet and the future of cyber war," Survival, vol. 53, no. 1, pp. 23–40, 2011

[28] P. P. Biswas, H. C. Tan, Q. Zhu, Y. Li, D. Mashima, and B. Chen, "A synthesized dataset for cybersecurity study of iec 61850 based substation," in 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids. IEEE, 2019, pp. 1–7

[29] scikit-learn. scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation. scikit-learn. https://scikit-learn.org/stable/

[30] TensorFlow. TensorFlow. https://www.tensorflow.org/