CP4101 Final Year Project (FYP)

CA Report

Teo Jia Wei

A0201915N

Project Title: Detect Cyber Attacks on Smart Grid System using Machine Learning

Project Objectives and Description:

This project focuses on the use of machine learning (ML) to detect cyber attacks, specifically malicious command injection, in the smart grid system. We focus on smart grid as a representative example of critical infrastructure. The smart grid systems consist of a number of network-connected, intelligent devices that monitor and control the power grid physical components, such as circuit breakers. As demonstrated in the Ukraine power plant cyber attack in 2015 [9], the injection of malicious control commands against such intelligent devices would cause a significant impact, e.g., massive power outage. One of the defense mechanisms against the attack of this sort is command authentication, which makes the context-aware decision on whether to execute the received remote control command or not. While some efforts using power system simulation have been made in the literature, they have some limitations, such as longer latency. Thus, in this project we explore feasibility and effectiveness of using machine learning technologies for the command authentication in smart grid.

The general objectives of this project primarily consist of:

1. Generation of training and testing datasets for ML models power flow simulator
2. Implementation of machine learning models for control command authentication in the smart grid.
3. Evaluation of machine learning models' performance for attack detection in smart grid

Ultimately, this project aims to design and implement an effective machine learning model to achieve the following:

1. Security: Detect malicious command injection attacks using ML algorithms
2. Accuracy: Achieve high accuracy with low false positive and false negative detection rates
3. Performance: Achieve low latency

Literature Review:

Smart grids are electric grids equipped with digital technology for remote monitoring of sensor data and control. The smart grid consists of two layers, cyber and physical systems [1]. The two layers are coupled with each other and form the cyber-physical environment. 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 grids to smart grids, 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, with Jurong Town Corporation developing one such grid for the Punggol Digital District [2].

Substations are critical entities in the power grid. Electricity from generators needs to be transformed through different voltage levels for efficient delivery to the end consumers and these transformations occur in substations [3]. Depending on the size of the services, there can be hundreds or thousands of substations, and within each substation, there can be many kinds of physical components, such as transformers, circuit breakers, shunt reactors, switchgear with busbars and so forth. For example, in Singapore, there are over 10,000 transmission/distribution substations under a single utility company [4].

Substations enable intelligent switching operations to facilitate automatic fault response and optimize power grid operations. Remote control and monitoring of substations are facilitated by communication network where the intelligent electronic devices (IED) act as the interface for the physical power system components. The measurement and status from the IEDs are input to the energy management system (EMS) for analysis, including power flow study, state estimation, etc., and subsequently management of the entire power system. Furthermore, desired remote control action of power system components, such as the circuit breakers, is channelised through the IEDs. As such, communication plays a vital role in conveying appropriate commands, measurements and statuses of the components for the management of the power system. Any intrusion in the communication channel with malicious intentions may jeopardise a part or whole of the power grid.

With the integration of computer networks and digital technology into the physical power grids, smart grid suffers from unauthorized access and its systems are now exposed to the same vulnerabilities that plague normal computer systems and networks [5]. Every asset of the Smart Grid, from home gateways to smart meters to substations to control rooms, is a potential target for a cyberattack.

According to attack targets, the cyber-attack against power grids can be classified into destroying the availability, integrity, confidentiality and authenticity of information in the grid. The availability destruction is embodied in unavailable information resulting from communication interruption, whose typical methods are denial-of-service (DOS) attack, black hole attack and attacks modifying network topology. The integrity destruction is embodied in incorrect information resulting from false data injection, whose typical methods are false data injection attack (FDIA), man-in-the-middle (MITM) attack and replay attack. The confidentiality destruction is embodied in data leakage and illegal usage, whose typical methods are brute force password cracking, utilisation of malware and internal employee attack [6]. The authenticity destruction is embodied in the injection of malicious control commands, whose typical methods are MITM and injection of rootkits to gain access to the system controls via privilege escalation. The threat of malicious commands will be heavily discussed in this literature as it is the focus of the project.

We briefly consider the problem of detection of false data injection (FDI) attacks, in which it is assumed that attackers can compromise the measurements of the grid. Such FDI attacks aim to affect the power system state estimation (PSSE) [7]. The state of a power system is generally defined as the voltage values on all the buses of the system. The modern power system, which is often operated near its operational limits for economic reasons, cannot be operated without a reliable PSSE. The impact of FDI attacks could range from economic consequences, through overloading and physical damage, to serious human hazard [8].

With a malicious command injection and 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 a power outage as shown in the infamous Ukraine power grid attack in 2015 [9]. Additionally, the aurora vulnerability showed that a cyberattack with only 30 lines of code can perform malicious commands which destroy physical components of the electric grid by rapidly opening and closing the generator's circuit breakers out of phase from the rest of the grid, ultimately causing it to explode [10].

In recent years, researchers have studied various possible cyber threat scenarios and their mitigation strategies on smart grids (e.g., [11], [12], [13]). The physical components in a smart substation communicate among themselves using standard industrial protocols including Modbus TCP, IEC 60870, IEC 61850, DNP3, etc. Several security technologies such as firewall, intrusion detection system (IDS), authentication, encryption, etc. have been advocated to protect the control network [14], [15]. Firewall and IDS typically work on a specific set of cyber rules to allow legitimate traffic inside a network. While firewall and (IDS) can flag/block unauthorized or malicious packets, they may fail to counter attacks that follow the normal communication model. Authentication and encryption would also fail if an authorised user mistakenly issues a harmful command to the system or if a legitimate device is compromised.

Cyber-based IDS helps monitor and analyze the network traffic in real-time. It identifies and sometimes blocks an event that does not satisfy the security policy of the system. Information exchange via designated protocols among network participants is often the key for network security policies. The cyber-based IDSes which include signature-based approach, behaviour-based approach, specification-based approach, etc. are common in detecting attacks in various cyber-physical systems including smart grids [16], [17]. However, such cyber-based approaches do not take up-to-date power grid system status for context-aware attack detection.

Furthermore, industrial control systems, such as those used in the Smart Electric Grid, are becoming more complex in their architecture and design. The Supervisory Control and Data Acquisition (SCADA) systems that are used are more interconnected and span multiple communication protocols and physical interfaces [18]. The methods by which data are collected from remote locations, as well as commercially available SCADA software developed for physically isolated systems, lead to more potential flaws in the hardware and software and provide a much larger attack surface to threat agents.

With the advent of artificial intelligence, machine learning has been used in many intruder detection systems as it provides low latency and high accuracy. Given a highly trained model, ML based detections often overpowers rules-based and physics based detection due to the high complexity of ML. With its promising computational and reasoning capabilities, several machine learning-based detectors and mitigation methods have been developed to deal with cyber threats.

The application of ML in IDS had been done in several contexts, e.g., in IoT based systems [27], for SCADA systems [28], and even for smart cities and related infrastructure [29]. The use of ML is especially prevalent in the detection of FDI attacks, with many recent research on it. Esmalifalak attempted to detect stealthy FDI attacks using a support vector machine (SVM) based technique and a statistical anomaly detection approach. They showed that SVM outperforms statistical approach when the model is trained with sufficient data [30]. Moreover, it can perform better than SVM and artificial neural network (ANN)-based detection mechanisms. Niu et al. presented a deep learning-based framework combining a convolutional neural network (CNN) and a long short term memory (LSTM) network to detect novel FDI attacks [33]. Yan et al. viewed the FDI attack detection problem as a binary classification problem and attempted to solve it using three different algorithms: SVM, K-nearest

neighbor (KNN), and extended nearest neighbor (ENN) [34]. Their experimental analysis showed that all these algorithms could be tuned to attain optimal performance against FDI attack detection. However, there was very limited finings on the application of ML against malicious commands injection.

The research work on defense against malicious commands attacks tries to differentiate between legitimate and illegitimate control commands. A semantic analysis framework based on cyber and physical characteristics of the smart grid was suggested in [22] to detect malicious commands. Similar hybrid (i.e., both cyber and physics rule-based) network intrusion detection systems to detect attacks on digital relays [23] and automated power distribution systems [24] were also studied. Remotely issued control commands to substations were authenticated by incorporating a delay [25] and using the latency to simulate the power system dynamics [26]. However, these works were often limited by high latency and focused on defense in the centralised grid system.

An intrusion detection method developed by my supervisors' team is ResiGate [35]. Currently, ResiGate focuses on the development of a physics-based intrusion detection for localized attacks, which happen within a single substation (or a cluster of nearby substations). ResiGate's physics-based approach could supplement the cyber-based rules widely employed for intrusion detection in the smart substation and grid. Specifically, ResiGate checks whether the execution of an incoming command violates any physical constraint (e.g., power flow limit on transmission lines) by using on-the-fly power flow simulation. Additionally, ResiGate augments the physics-based bad data detection with machine learning to enhance and speed-up the performance of the online false data detection. ResiGate uses gradient boosting tree (GBT) to scan data and any data flagged as suspicious will be further validated by the existing physics-based analysis using state estimation. ResiGate is the first security appliance solution to be implemented in distributed, standalone manner in the field substation systems. Additionally, ResiGate is also the first system that combines the efficiency of an ML-based approach with the accuracy of real time physics-based analysis for the practical deployment in field substations, and implements with a fully-functioning research prototype.

However, when checking for malicious commands and authenticating commands, ResiGate suffers from high latency as power flow simulations are run each time a command is made. This latency can be longer than 1s, which greatly affects the performance of the substations, since the substations are continuously receiving new commands all the time in a dynamic environment.

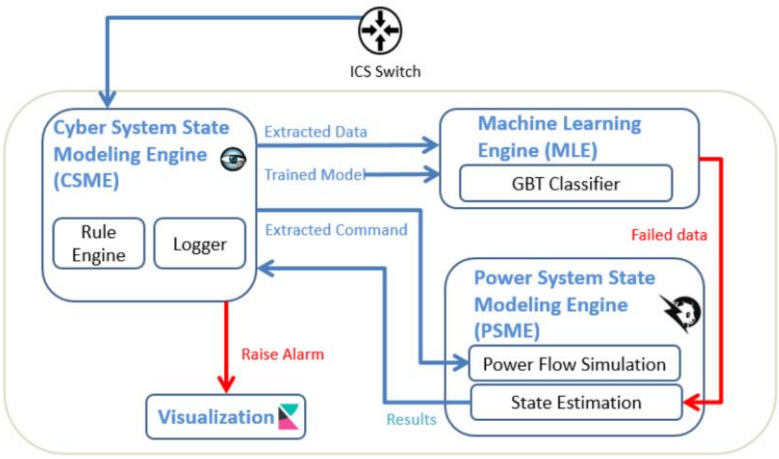

Fig 1: ResiGate's Architecture

As such, in this project, the focus will be on the development of a machine learning based intrusion detection for localized malicious commands attacks, which happen within a single substation (or a cluster of nearby substations). Through this work, we hope to develop a fully functioning research prototype that can use machine learning to validate the legitimacy of a command.

Lastly, in order 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 they can be generated synthetically. The generation of data synthetically can be done using rule-based or ML based. According to the existing research works, four types of ML algorithms are utilized to generate malicious data to launch an attack in the smart grid. K-means and Q-learning algorithms are used in [36], [37] to launch the line switching attacks, which are a form of malicious commands injection. In contrast, Q-learning, generative adversarial networks (GAN), and linear regression (LR) models are used to generate false data injection (FDI) attacks [38], [39]. Ahmadian et al. presented a GAN-based false load data generator in [40]. The attacker's goal was to maximize the generation cost by injecting that false load data into the system. Recently, Chen et al. also presented a Q-learning based FDI attack generator against the automatic voltage control using partially observable Markov decision process and was able to create a voltage sag on IEEE 39 bus system [41].

Motivation:

Cyber security was not a major concern 2 decades ago when non-digital end-to-end copper cables were still widely used in substation communication and "security through obscurity" worked well as there were not many studies or cyber attack trials reported. However, with the transition of traditional non-digital power grids to smart grids, the introduction of computer networks and technology opened up a wide spectrum of attack surfaces that can disrupt or destroy our power supply with a single click.Furthermore, electrical substations are being increasingly digitalized due to the implementation of smart grids, they become more prone to cyber attacks. A single cyber attack can potentially lead to disastrous outcomes such as widespread blackouts and economical losses. Such disastrous impacts can be seen in the infamous Ukraine Power Grid Cyber Attack.

As such, there are many studies and research done to prevent cyber attacks on smart grids from happening. These studies range from physics-based approaches to the more recent machine learning approaches. While the detection mechanisms for detecting fake measurement data as well as malicious control commands injected by attackers are discussed in the literature [42], [43], [44], they are either fully or partially centralized for handling heavy computation for physics-based analysis, whose communication path may become an additional attack surface to either delay or disable the processing. Centralized solutions suffer from a lack of visibility and are thus incapable of detecting localized attacks. There is a need to develop detection in these localized areas such as substations, as substations are critical entities in the power grid and an intrusion into a substation can not only affect the region directly connected to that substation, but also lead to cascading events and eventually catastrophic failure of the grid. This is especially so if an attack can remotely compromise multiple substations in a coordinate manner to launch attacks. Furthermore, a centralized scheme is unable to block malicious attacks injected directly into the substations.

While statistical-based and physics-based approaches for both bad data detection and command validation is usually accurate, it tends to be computationally expensive, and may not complete in time to allow intervening actions such as switch opening. They suffer from high latency, especially during transient stability analysis and these approaches become inaccurate when there are insufficient data.

As such, machine learning serves as a potential candidate for faster preliminary detection. Due to its computational speed and robustness, ML implementations can perform with lower latency and potentially higher accuracy in detection. ML models have been widely researched in the field of FDIA, with ResiGate being an example. However, there is limited coverage of ML in the field of command authentication and detecting malicious commands. Lastly, to train and test our machine learning models, we need realistic attack data which is often not readily available. To overcome this, synthetic generation of datasets for the ML models can be achieved using power flow simuluator.

Problem Statement:

Given the importance of command authentication and substations being a critical component of the power grid, as well as the computational abilities and accuracy that machine learning provides, how can we implement state-of-the-art machine learning algorithms to detect malicious commands injected into substations in smart grids with high accuracy and low latency? Furthermore, how can we generate synthetic datasets to train and test our machine learning models that can effectively detect malicious control commands under real-world attack scenarios?

Limitations and Assumptions:

Whilst researching on the project, there are a few limitations that I faced. Firstly, the power grid used in my research and the existing ResiGate's work is a 3-substation model, which is considered a small system. Hence, the results may not scale well when applied to a larger or more complex system. To overcome this, there may be plans to test the on larger systems such as a 9-bus system configuration. Secondly, we assume that when the ML model is online, there is no maintenance work being done to the substations such as the disconnection of any substations. This is because the ML model will detect this disconnection as an attack, leading to a false positive. As such, we assume that the operator of the substations will disable the ML detection when maintenance work is being carried out. Additionally, we assume that there are no faults in the system and commands are always given remotely. With these assumptions laid out, the commands given to the substations can only be either valid remote commands by the operators or malicious commands injected by the attacker, allowing us to evaluate the ML model fairly and accurately.

Research Methodology:

The research methodology of this project mainly comprises of 3 stages: initial research, continuous improvement of ML model, and implementation of the best ML model in an intruder detection system (IDS). Most work will be done in continuously experimenting with different ML models and training them with different training datasets, before testing them on a variety of attack scenarios. Through countless of iterations and improvements, the best performing ML model will be selected and implemented in an existing IDS.

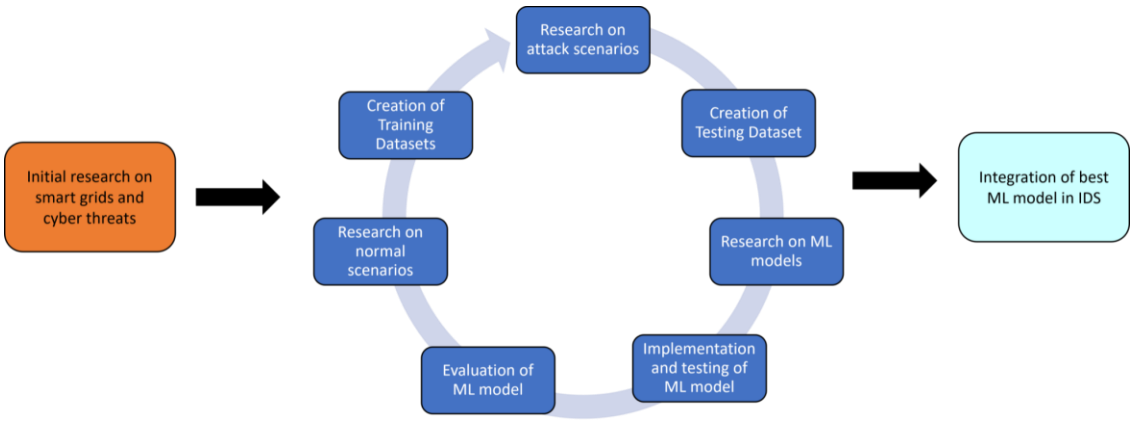

Fig 2: Research Methodology Flowchart

Management Tools:

To ensure an effective and efficient workflow over the semester, proper planning for project and resource management is done. To track my weekly progress, I made a list of tasks to be done at the start of every week and ensured that these tasks are done by the end of the week. I also synced up with my supervisors every week through zoom to consult and update my current progress. Informal discussions and updates are also done through skype chats. Additionally, I used Github to track and store my codes. This allowed me to revert to any old versions of the code if there is a need to. Lastly, Google Drive is used to store any reports and results in the cloud. These documents can be accessed easily by my supervisors and it serves as a backup.

For resource and time management, I allocated around 16h a week spent on this project to ensure I meet my deliverables for the week. The time is spent on planning, designing algorithms, implementing datasets and ML models, as well as testing and evaluating the ML models.

Current Progress:

Building fundamentals:

To kickstart my final year project, the initial tasks wereto familiarize myself with smart grids, substations and the existing works on the cyber threats and protection. To gain domain knowledge, I read my research's team's existing work, namely ResiGate, which attempts to block attacks in the form of bad data injection and malicious command injection using physics-based simulation. Additionally, I consulted my supervisors Dr Mashima and Dr Biswas whenever I have questions and search for reference materials online to enhance my understanding when necessary. Doing literature reviews boosted my understanding of the smart grid and its cyber threats as well. Additionally, I familiarized and refreshed myself with Python libraries that will be useful, namely Pandapower [45] (for power flow simulation), Scikit-learn [46] and Tensorflow [47] (for machine learning).

Datasets:

After getting the fundamentals laid out, I began implementing different datasets and testing out different ML models. Using Pandapower, I created a 3-substation Power Grid to simulate the environment. There are a total of 34 switches (denoted by CB-x), 6 transformers (denoted by Tx), 18 p and q loads for each line (denoted by Lines Lx) and 2 Feeders (denoted by Feeder x). The initial total load is also calculated as well. As such, there is a total of 79 features in the dataset, with another column for its label (binary classification) and another column for the reason of a violation if any. There are different types of violations, namely: grid config violation, lines/transformer overloading, bus voltage overload, total load lost percentage and number of open switches. A violation of the power grid will be considered as an anomalous data point in our dataset. A classification of 1 represents an anomaly while a classification of 0 represents a normal datapoint.

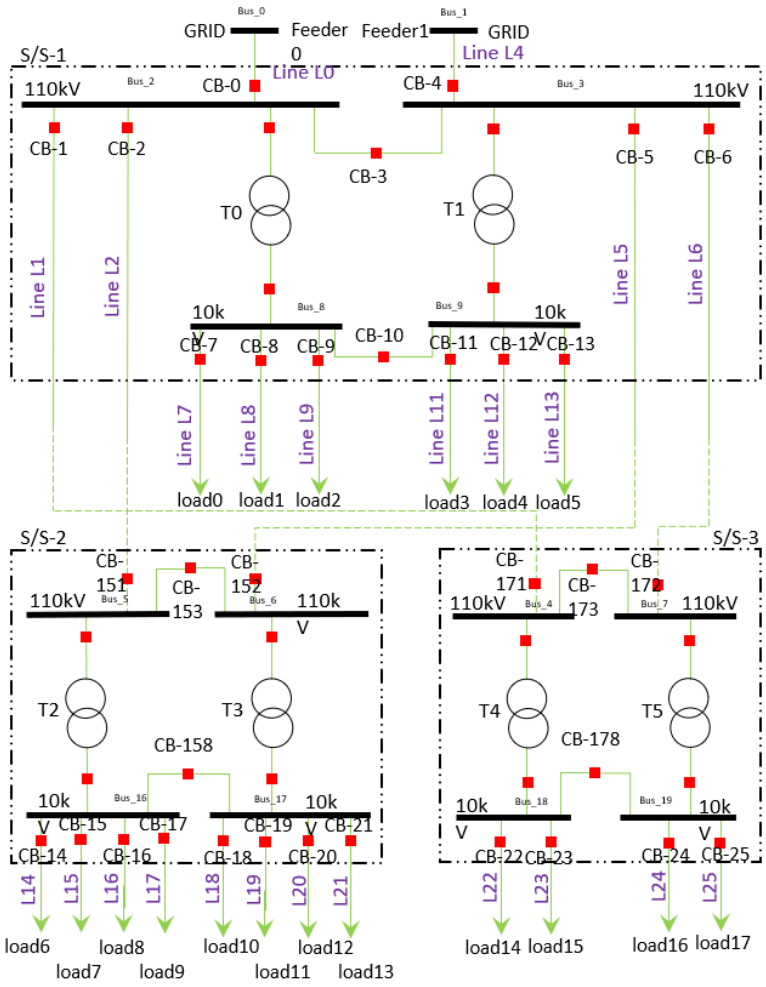

Fig 3: 3-Substation Grid Configuration

Training dataset:

Training data primarily contains valid and realistic states of the substations. We can imagine it to be taking a snapshot of the state of the power grid at a certain point and saving it as a datapoint. For most training datasets, we start with a valid configuration of the power grid with no anomaly before adding

different constraints to it, to simulate some form of attack. We explore the constrained, Ncombined, Unique Normal, and UNOR datasets.

Constrained dataset:

For constrained, we ensure that for each substation, at least 1 feeder/transformer is online and always connected. This results in a valid configuration. This means that the transformer/feeder bus couplers (CB-3/10/153/158/173/178) will be closed if needed to ensure all loads are connected in the grid. Afterward, we randomly switch on/off the switches connecting to the loads (load0 – load17). This dataset consists of data points that are of valid grid configurations but may violate other conditions.

NCombined dataset:

The constrained datasets provided a simple representation of the grid. However, these data points are independent of one another and do not represent the real-world context of a smart grid and the substations. As the substations are in a continuous and dynamic environment, the switches/loads are constantly changing. As such, to simulate a more realistic training dataset, there is a need to generate a valid starting point for any grid configuration, before applying options to modify the grid. This results in a "mini" dataset for each starting data point. We apply one "option" at a time, and the resultant modification of the grid constitutes a new data point. "Options" are possible ways in which to make the grid incrementally approach states that violate power flow requirements (e.g. connecting a load, disconnecting a feeder, disconnecting a generator). Options are continuously applied till we reached a violation or run out of options. Hence, each "mini" dataset will consist of the starting datapoint followed by the datapoints with options sequentially applied.

This gives rise to the Ncombined dataset (Normal with disturbance + Normal without disturbance). The starting data point can be any data point with all its switches being closed (Normal without disturbance) or any data point that has a valid grid configuration (Normal with disturbance), similar to the constrained dataset.

Unique Normal dataset:

The Unique Normal dataset is an improved variant of the Ncombined dataset. It provides a wider coverage of different grid permutations as well as load profiles, allowing the ML models to train better. Instead of having any random valid configuration as a starting point for our "mini" dataset, we permutate all possible unique valid configurations. This means that the transformer/feeder bus couplers (CB-3/10/153/158/173/178) will be closed if needed to ensure all loads are connected in the grid. For example, an invalid configuration which is a violation occurs when CB-3 is open with either Feeder 0 being open. This is because there will be disconnection and load lost to lines L7-L9. In summary, at every transformer/feeder bus couplers, we need at least 2 out of 3 switches closed. This results in a total of 4096 valid configurations (4 possible permutations of 2/3 switches closed ^ 6 transformer/feeder bus coupler).

For each 4096 valid configurations, we generate 5 different load profiles for the grid as our starting point from 0%-40%, 40%-70%, 70%-100%, 100%-120% and 120%-150%. This is to diversify the training datasets with varying load profiles. Hence, we will have 4096*5 different starting points for our "mini" datasets. We then proceed to apply the options sequentially, similar to the Ncombined dataset to generate the rest of our datapoints.

Unique Normal Options Random (UNOR) dataset:

Lastly, the Unique Normal Options Random (UNOR) dataset is an extension to the Unique Normal dataset. Essentially, UNOR adds an additional step after the generation of Unique Normal dataset: After applying an option, we randomly switch on or off any feeder/transformer or load switches. This is to ensure our dataset also contains some invalid configuration for our machine learning models to learn.

	Constrained	NCombined	Unique Normal	Unique Normal Options Random
Config Violation: Trafo1	0	0	0	1294
Config Violation: Trafo2	0	0	0	1686
Config Violation: Trafo3	0	0	0	1671
Config Violation: Feeder1	0	239	5	128
Config Violation: Feeder2	0	208	567	1034
Config Violation: Feeder3	0	239	589	953
Lines over 80%	7914	582	2016	2445
Load Lost > 20%	0	1783	5442	1772
Transformers over 80%	1428	275	5653	5624
More than 8 open switches	3247	699	2176	407
bus PU violated	0	0	140	170
No anomaly	7411	15975	59953	28497
Total	20000	20000	76541	45681

Figure 4: Breakdown of Training Datasets

Testing datasets:

Testing datasets primarily consists of datapoints that violate the power grid conditions and each datapoint provides a realistic state of a power grid that has been modified by a malicious command injection.

Unconstrained Dataset:

The motivation for unconstrained is simple: randomly switching on and off any switches, transformers and feeders to simulate a random grid environment for training data. This simulates an attacker malicious disrupting the grid by toggling on/off random switches to cause confusion.

Special Dataset:

Special dataset is similar to constrained dataset in that datapoints always ensure connection between at least 1 feeder and all non-load buses. However, datapoints are generated as sets, consisting of 1 normal datapoint where substations are operating safely, followed by anomalous datapoints where substations violate power flow restrictions. For each set, the grid is initialized in a state where all feeders and generators are online and connected, and all loads are disconnected. Datapoints are generated from initial state by repeatedly applying options. Normal datapoint is created by applying a fixed number of options, then anomalous datapoints in same set are created by incrementally applying options from the normal datapoint until a power flow restriction is violated.

Attack Dataset:

Another form of malicious command will be switching on/off switches such that we will have a violation to the power grid configuration. This can occur when there are more than 2 switches being open at any of the transformer/feeder bus couplers as explained in the Unique Normal training dataset. As such, we can generate such a dataset by initializing all possible grid configurations as our testing dataset. At each transformer/feeder bus coupler, we have 8 possible permutations of the 3 switches being closed and open (2^3 permutations). As we have 6 of these transformer/feeder bus coupler, we will create 8^6 = 262144 possible unique grid configurations. Out of 262144, there will be a total of 262144-4096 = 258048 violations. This dataset simulates the idea of an attacker maliciously opening switches such that we will have an invalid power grid configuration state.

	Unconstrained	Attack	Special
Config Violation: Trafo1	5002	65536	0
Config Violation: Trafo2	586	16384	0
Config Violation: Trafo3	83	4096	0
Config Violation: Feeder1	10112	131072	0
Config Violation: Feeder2	3705	32768	0
Config Violation: Feeder3	446	8192	0
Lines over 80%	6	0	8361
Load Lost > 20%	0	0	0
Transformers over 80%	0	0	6639
More than 8 open switches	56	0	0
bus PU violated	1	0	0
No anomaly	3	4096	5000
Total	20000	262144	20000

Figure 5: Breakdown of Testing Datasets

Flowcharts and pseudocodes of the generation of these training and testing datasets can be found in the appendix section.

Machine Learning Models:

Using the datasets generated, I tested the effectiveness of various classification machine learning models, namely Logistic Regression, Decision Tree, Random Forest, Adaboost, XGBoost. This is done using the ML libraries scikit-learn. For each training dataset, we trained 5 different ML models using it and tested them with the 3 different attack datasets. Additionally, grid search is performed on the ML models to find the best hyperparameters. As such, we will have a total of 4 training datasets * 5 ML models * 3 testing datasets = 60 results. Due to brevity's sake, full results are shown in the appendix section.

When evaluating the results, we placed more emphasis on false negatives. This is because if a malicious command is treated as a normal datapoint by the ML model, the malicious command 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 in checking the validity of the command, as any command

treated as an anomaly will be passed to the power flow simulator estimator to be checked again. Hence, false negatives are costlier than false positives and we aim to reduce or eliminate false negatives.

Trained using UNOR, Tested on Attack
+VE: 258048, -VE: 4096

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	3133	196758	61290	963	76.25%	0.9951	0.7624	0.8634
Decision Tree	4096	249530	8518	0	96.75%	1	0.9669	0.9832
Random Forest	4096	253795	4252	0	98.37%	1	0.9835	0.9916
AdaBoost	4096	249530	8518	0	96.75%	1	0.9669	0.9668
XGBoost	4096	256456	1592	0	99.39%	1	0.9938	0.9969

Fig 6: Results of ML models trained with UNOR dataset and tested on Attack dataset

Trained using UNOR, Tested on Special
+VE: 15000, -VE: 5000

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	4959	7492	7508	41	62.25%	0.9945	0.4994	0.6649
Decision Tree	4346	10943	4057	654	76.44%	0.9436	0.7295	0.8228
Random Forest	4949	14477	523	51	97.13%	0.9964	0.9651	0.9805
AdaBoost	5000	6657	8343	0	58.28%	1	0.4438	0.6147
XGBoost	4966	14483	517	34	97.24%	0.9976	0.9655	0.9813

Fig 7: Results of ML models trained with UNOR dataset and tested on Special dataset

In terms of model performance, XGBoost showed the most promising results across training with 4 different training datasets, with the highest accuracy and recall consistently. Using UNOR training dataset and attack testing dataset as a comparison, we see that XGBoost is able to identify the greatest number of true positives and have the least number of false negatives as compared to the other ML models. This can be seen when tested against the both the attack and special datasets. Random Forest also performed relatively well as compared to the other models, with logistic regression performing the worst, due to its low model complexity.

Trained using XGBoost, Tested on Attack
+VE: 258048, -VE: 4096

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Constrained	0	258048	0	4096	98.43%	0.9843	1	0.9921
NCombined	0	258048	0	4096	98.43%	0.9843	1	0.9921
Unique Normal	4096	249530	8518	0	96.75%	1	0.9669	0.9668
UNOR	4096	256456	1592	0	99.39%	1	0.9938	0.9969

Fig 8: Results of XGBoost Models trained using different training datasets and tested on Attack dataset

In terms of training dataset, models trained with Constrained and Ncombined performed the best in terms of recall. Using XGBoost as the ML model, the datasets have the highest recall of 1 when tested with the attack dataset. However, this is because they failed to detect all of the true negatives datapoints and labelled them as false positives instead. This may imply that the models are incompetent in differentiating between positive or negative data points and are instead treating all data points as positive data points. On the other hand, whilst the UNOR dataset has a lower recall than the constrained dataset, it can predict all of the true negatives data points correctly, leading to higher accuracy.

Other than ski-learn models, I experimented with Artificial Neural Networks (ANN) model using TensorFlow. To find the best loss function, activation function, number of layers and nodes in each layer, grid search is performed using the UNOR training dataset. The best parameters are as shown:

Number of Layers	4
First Layer Nodes	256
Second Layer Nodes	172
Third Layer Nodes	88
Fourth Layer Nodes	1
Activation Function	Relu (Sigmoid at the last layer)
Loss Function	Binary Cross Entropy
Epoch	100
Batch Size	100

The ANN model is then evaluated against the different testing datasets, with promising results when tested with the Attack and Special dataset, outperforming models trained using XGBoost.

Trained using UNOR, Tested on Unconstrained, Attack and Special

Test Dataset	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Unconstrained	2	19117	880	1	0.956	0.9999	0.956	0.9775
Attack	4096	255903	2145	0	0.9918	1	0.9917	0.9958
Special	4954	14846	154	46	0.99	0.9908	0.9699	0.9802

Fig 9: Results of ANN Model trained using UNOR dataset and tested on different testing datasets

Research Plan for the next semester:

For the next semester, I plan to study the feature importance of each feature in the dataset in detail. Currently, using XGBoost's in-built get_booster() and get_score() functions, there are preliminary findings on the feature importance in the dataset. The "weight" of the feature is used as the basis for comparison, with "weight" being the number of times a feature is used to split the data across all trees. Load 13 is shown to have the highest feature importance, with the other features such as load 2, total initial load and load 4 having similar importance with one another.

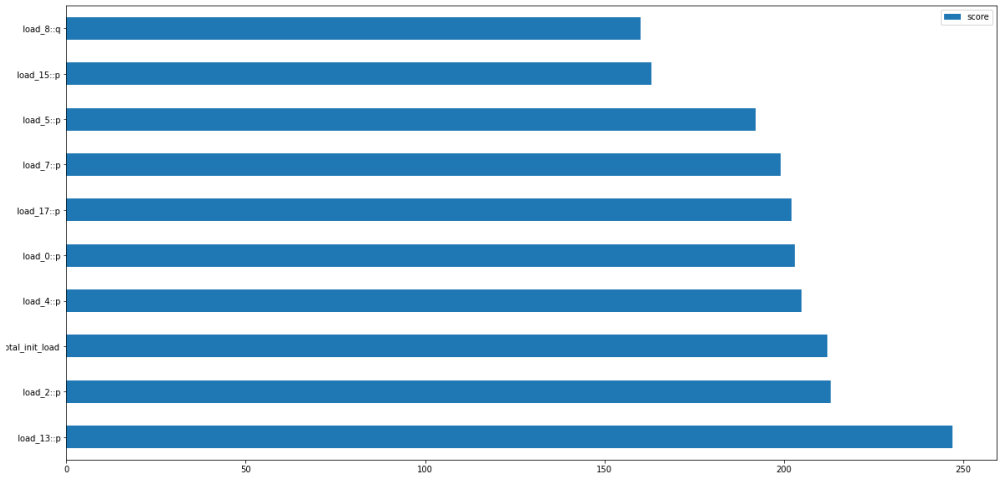

Fig 10: Top 10 features based on feature importance using XGBoost

However, the results are still inconclusive and there is no clear explanation for load 13 being the most important feature as it appears to be a normal load in the grid. More research will be done in the next semester to better understand the feature importance of the datasets.

Additionally, I plan to generate more test datasets that encompass different attack scenarios to test the effectiveness of the ML models against a variety of attack scenarios. An example of an attack scenario will be having malicious commands switching off switches one at a time till the power grid configuration is violated.

There may also be further exploration of other potential ML models, such as RNN or even reinforcement learning (Q-learning). Imposing heavier costs or weights on false negatives to increase recall rate will be explored as well. Lastly, the best performing ML model will be chosen to be implemented into an IDS system that can detect actual commands in a smart grid system. This can be done using Zeek [48], a network scripting language and security monitoring tool and Modbus which is a smart grid communication protocol [49].

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] Soh, S. B. (n.d.). *Singapore's Electricity Industry* https://www.iitk.ac.in/ime/anoops/FoR-15-singapore/photos/PPTs/Day%20-%201%20Singapore/Mr.%20Sai%20Bor%20Soh%20-%20Power%20Sector%20Regulation%20in%20Singapore.pdf

[5] Faquir, D., Chouliaras, N., Sofia, V., Olga, K., & Maglaras, L. (n.d.). *Cybersecurity in smart grids, challenges and solutions*. AIMS Electronics and Electrical Engineering. Retrieved April 2, 2022, from https://www.aimspress.com/article/doi/10.3934/electreng.2021002?viewType=HTML

[6] Q, W., W, T., & T, T. (2018, October 18). *Review of the false data injection ... - wiley online library*. Review of the false data injection attack against the cyber-physical power system. Retrieved April 2, 2022, from https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/iet-cps.2018.5022

[7] Deng, R. long. (2017, April 2). *False data injection on state estimation in power systems*. Retrieved April 2, 2022, from https://www.cs.unb.ca/~rlu1/paper/DengXLLV17.pdf

[8] Drayer, E., & Routtenberg, T. (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

[9] Zetter, K. (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/

[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] A. Gupta, A. Anpalagan, G. H. Carvalho, L. Guan, and I. Woungang, "Prevailing and emerging cyber threats and security practices in iotenabled smart grids: A survey," Journal of Network and Computer Applications, vol. 132, pp. 118–148, 2019.

[12] H. He and J. Yan, "Cyber-physical attacks and defences in the smart grid: a survey," IET Cyber-Physical Systems: Theory & Applications, vol. 1, no. 1, pp. 13–27, 2016.

[13] H. C. Tan, C. Cheh, B. Chen, and D. Mashima, “Tabulating cybersecurity solutions for electrical substations towards pragmatic design and planning,” in presented at the IEEE Innovative Smart Grid Technologies Asia (ISGT Asia), Chengdu, China. IEEE, 2019.

[14] M. Z. Gunduz and R. Das, “Cyber-security on smart grid: Threats and potential solutions,” Computer Networks, p. 107094, 2020.

[15] J. Cai, Y. Zheng, and Z. Zhou, “Review of cyber-security challenges and measures in smart substation,” in 2016 International Conference on Smart Grid and Clean Energy Technologies. IEEE, 2016, pp. 65–69.

[16] [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.

[17] R. Berthier and W. H. Sanders, “Specification-based intrusion detection for advanced metering infrastructures,” in 2011 IEEE 17th Pacific Rim International Symposium on Dependable Computing. IEEE, 2011, pp. 184–193.

[18] Barr, D. (2004, October). *NISCC good practice guide on SCADA firewalls V1-4*. National Communications System . Retrieved April 2, 2022, from https://icscsi.org/library/Documents/Best_Practices/NISCC%20-%20Good%20Practice%20Guide%20on%20Firewall%20Deployment%20for%20Control%20Systems.pdf

[19] Y. Liu, P. Ning, and M. Reiter, “False data injection attacks against state estimation in electric power grids,” ACM Transactions on Information and System Security (TISSEC), vol. 14, no. 1, pp. 1–33, 2011.

[20] G. Dan and H. Sandberg, “Stealth attacks and protection schemes for state estimators in power systems,” in 2010 first IEEE international conference on smart grid communications. IEEE, 2010, pp. 214–219.

[21] 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

[22] H. Lin, A. Slagell, Z. Kalbarczyk, P. W. Sauer, and R. K. Iyer, “Semantic security analysis of SCADA networks to detect malicious control commands in power grids,” in Proceedings of the first ACM workshop on Smart energy grid security, 2013, pp. 29–34.

[23] G. Koutsandria, V. Muthukumar, M. Parvania, S. Peisert, C. McParland, and A. Scaglione, “A hybrid network IDS for protective digital relays in the power transmission grid,” in 2014 IEEE International Conference on Smart Grid Communications. IEEE, 2014, pp. 908–913.

[24] M. Parvania, G. Koutsandria, V. Muthukumary, S. Peisert, C. McParland, and A. Scaglione, “Hybrid control network intrusion detection systems for automated power distribution systems,” in 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE, 2014, pp. 774–779.

[25] D. Mashima, P. Gunathilaka, and B. Chen, “Artificial command delaying for secure substation remote control: Design and implementation,” IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 471–482, 2017.

[26] D. Mashima, B. Chen, T. Zhou, R. Rajendran, and B. Sikdar, “Securing substations through command authentication using on-the-fly simulation of power system dynamics,” in 2018 IEEE International

Conference on Communications, Control, and Computing Technologies for Smart Grids. IEEE, 2018, pp. 1–7.

[27] 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.

[28] 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.

[29] 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.

[30] Mohammad Esmalifalak, Lanchao Liu, Nam Nguyen, Rong Zheng, and Zhu Han. Detecting stealthy false data injection using machine learning in smart grid. IEEE Systems Journal, 11(3):1644–1652, 2014.

[31] Youbiao He, Gihan J Mendis, and Jin 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.

[32] Hadis Karimipour, Ali Dehghantanha, Reza M Parizi, Kim-Kwang Raymond Choo, and Henry Leung. A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids. IEEE Access, 7:80778–80788, 2019

[33] Xiangyu Niu, Jiangnan Li, Jinyuan Sun, and Kevin Tomsovic. Dynamic detection of false data injection attack in smart grid using deep learning. In 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), pages 1–6. IEEE, 2019.

[34] Jun Yan, Bo Tang, and Haibo He. Detection of false data attacks in smart grid with supervised learning. In 2016 International Joint Conference on Neural Networks (IJCNN), pages 1395–1402. IEEE, 2016.

[35] Biswas, P. P., Mohanraj, V., Sourav, S., Chen, B., & Mashima, D. (2021). Intrusion Detection for Electrical Substations: Machine Learning Meets Physics-based Analysis. *Under submission*.

[36] Shuva Paul, Md Rashedul Haq, Avijit Das, and Zhen 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.

[37] Zhen Ni, Shuva Paul, Xiangnan Zhong, and Qinglai 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.

[38] Zhen Ni and Shuva Paul. A multistage game in smart grid security: A reinforcement learning solution. IEEE transactions on neural networks and learning systems, 30(9):2684–2695, 2019

[39] Rehan Nawaz, Muhammad Awais Shahid, Ijaz Mansoor Qureshi, and Muhammad Habib Mehmood. Machine learning based false data injection in smart grid. In 2018 1st International Conference on Power, Energy and Smart Grid (ICPESG), pages 1–6. IEEE, 2018.

[40] Saeed Ahmadian, Heidar Malki, and Zhu Han. Cyber attacks on smart energy grids using generative adverserial networks. In 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), pages 942–946. IEEE, 2018.

[41] Ying Chen, Shaowei Huang, Feng Liu, Zhisheng Wang, and Xinwei Sun. Evaluation of reinforcement learning-based false data injection attack to automatic voltage control. IEEE Transactions on Smart Grid, 10(2):2158–2169, 2018.

[42] F. C. Schweppe and J. Wildes, "Power system static-state estimation, Part I: Exact model," IEEE Transactions on Power Apparatus and systems, no. 1, pp. 120–125, 1970.

[43] J. Chen and A. Abur, "Placement of pmus to enable bad data detection in state estimation," IEEE Transactions on Power Systems, vol. 21, no. 4, pp. 1608–1615, 2006.

[44] D. Mashima, B. Chen, T. Zhou, R. Rajendran, and B. Sikdar, "Securing substations through command authentication using on-the-fly simulation of power system dynamics," in 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids. IEEE, 2018, pp. 1–7.

[45] Pandapower. (n.d.). pandapower. http://www.pandapower.org/

[46] Scikit-learn. (n.d.). *scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation*. scikit-learn. https://scikit-learn.org/stable/

[47] TensorFlow. (n.d.). TensorFlow. https://www.tensorflow.org/

[48] Zeek. (n.d.). *The Zeek Network Security Monitor*. Zeek. https://zeek.org/

[49] PyModbus. (n.d.). *PyModbus - A Python Modbus Stack.* https://pymodbus.readthedocs.io/en/3.0.0/readme.html

Appendix:

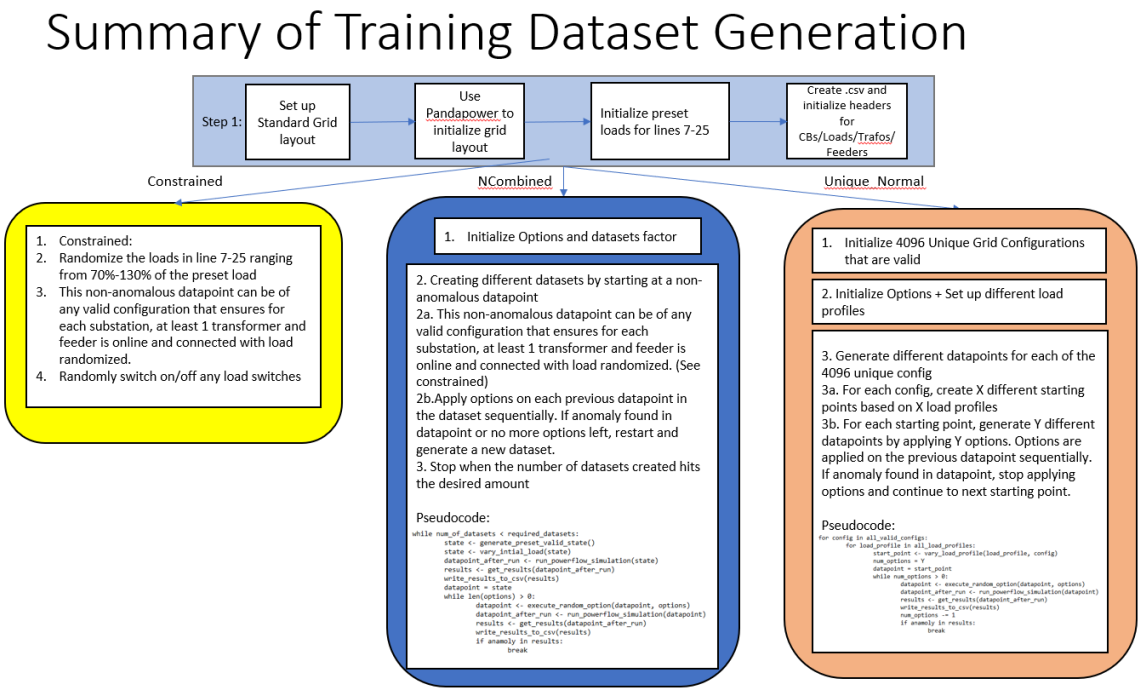

Figure 11: Flowchart of the different training datasets

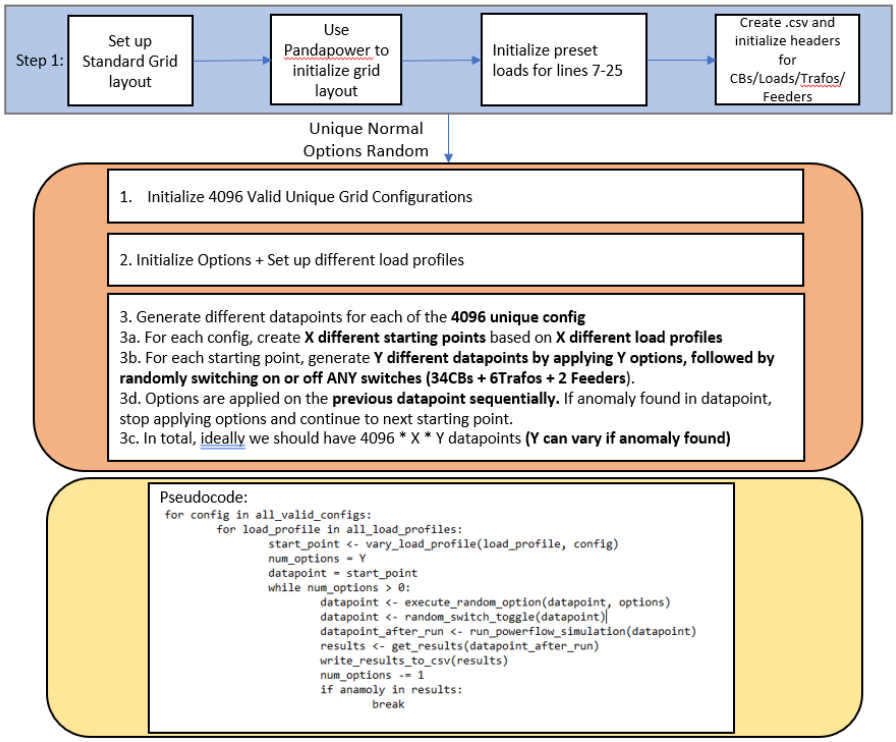

Figure 12: Flowchart of UNOR dataset generation

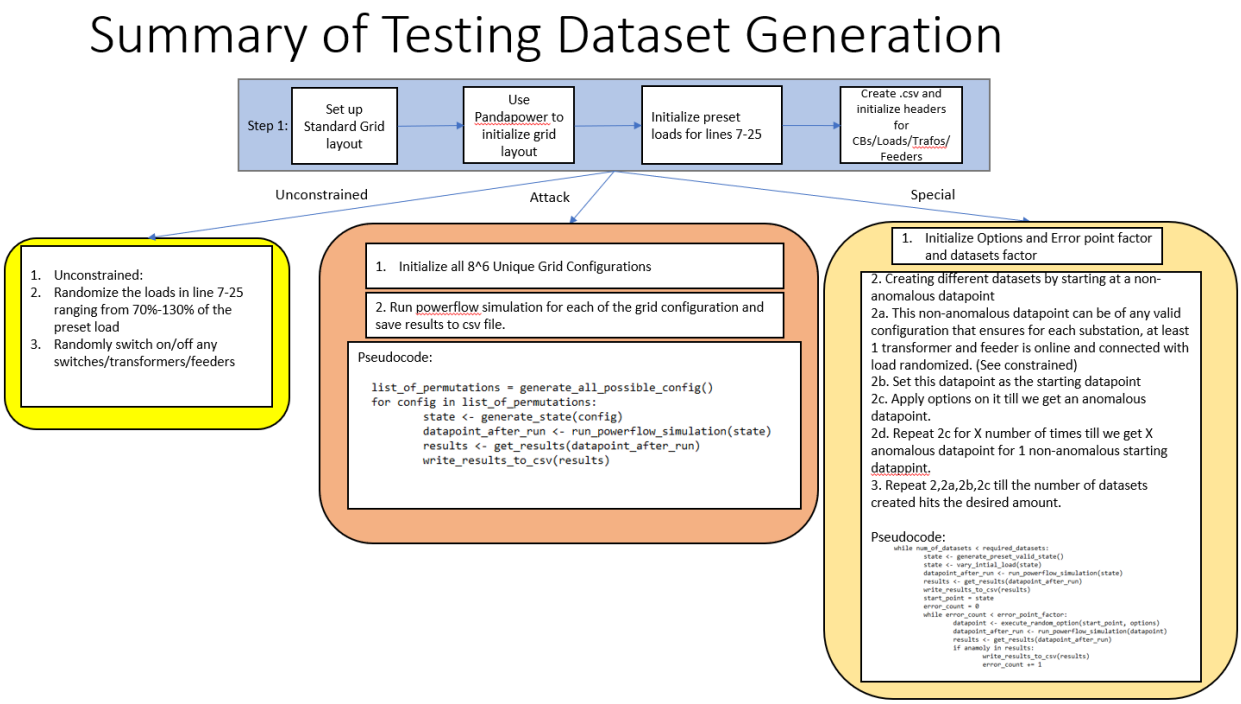

Figure 13: Flowchart of the different testing datasets

ML Results

Trained using Constrained, Tested on Unconstrained
+VE: 19997, -VE: 3

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	3	5479	14518	0	27.41%	1	0.2739	0.4301
Decision Tree	2	19873	124	1	99.37%	0.9999	0.9937	0.9968
Random Forest	2	19990	7	1	99.96%	0.9999	0.9996	0.9997
AdaBoost	2	19987	10	1	99.94%	0.9999	0.9994	0.9997
XGBoost	2	19987	10	1	99.94%	0.99994	0.9994	0.9997

Fig 14: Results of ML models trained with Constrained dataset and tested on Unconstrained dataset

ML Results

Trained using NCombined, Tested on Unconstrained
+VE: 19997, -VE: 3

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	3	3774	16223	0	18.88%	1	0.1887	0.3175
Decision Tree	2	19707	290	1	98.54%	0.9994	0.9854	0.9926
Random Forest	0	19991	6	3	99.95%	0.9998	0.9996	0.9997
AdaBoost	0	19996	0	3	99.98%	0.9998	1	0.9999
XGBoost	0	19997	0	3	99.98%	0.9998	1	0.9999

Fig 15: Results of ML models trained with Ncombined dataset and tested on Unconstrained dataset

ML Results

Trained using Unique Normal, Tested on Unconstrained
+VE: 19997, -VE: 3

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	2	18702	1295	1	93.52%	0.9999	0.9350	0.9665
Decision Tree	2	19439	558	1	97.20%	0.9999	0.9720	0.9858
Random Forest	2	199983	14	1	99.92%	0.9994	0.9992	0.9996
AdaBoost	1	19994	3	2	99.97%	0.9998	0.9998	0.9998
XGBoost	0	19997	0	3	99.98%	0.9998	1	0.9999

Fig 16: Results of ML models trained with Unique Normal dataset and tested on Unconstrained dataset

ML Results

Trained using UNOR, Tested on Unconstrained
+VE: 19997, -VE: 3

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	3	11217	8780	0	56.10%	1	0.5609	0.7187
Decision Tree	1	19868	129	2	99.34%	0.9998	0.9935	0.9967
Random Forest	0	19996	1	3	99.98%	0.9998	0.9999	0.9998
AdaBoost	0	19996	1	3	99.98%	0.9998	0.9999	0.9998
XGBoost	0	19997	0	3	99.98%	0.9998	1	0.9999

Fig 17: Results of ML models trained with UNOR dataset and tested on Unconstrained dataset

ML Results

Trained using Constrained, Tested on Attack
+VE: 258048, -VE: 4096

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	4080	1895	256153	16	2.27%	0.9916	0.007345	0.01457
Decision Tree	0	258048	0	4096	98.43%	0.9843	1	0.9921
Random Forest	0	258048	0	4096	98.43%	0.9843	1	0.9921
AdaBoost	0	258048	0	4096	98.43%	0.9843	1	0.9921
XGBoost	0	258048	0	4096	98.43%	0.9843	1	0.9921

Fig 18: Results of ML models trained with Constrained dataset and tested on Attack dataset

ML Results

Trained using NCombined, Tested on Attack
+VE: 258048, -VE: 4096

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	4096	61160	196888	0	24.89%	1	0.2370	0.3831
Decision Tree	3072	235144	22904	1024	90.87%	0.9956	0.9112	0.9515
Random Forest	0	258048	0	4096	98.43%	0.9843	1	0.9921
AdaBoost	0	258048	0	4096	98.43%	0.9843	1	0.9921
XGBoost	0	258048	0	4096	98.43%	0.9843	1	0.9921

Fig 19: Results of ML models trained with NCombined dataset and tested on Attack dataset

ML Results

Trained using Unique Normal, Tested on Attack
+VE: 258048, -VE: 4096

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	4096	4844	253204	0	3.41%	1	0.01877	0.03685
Decision Tree	4096	206319	51729	0	80.26%	1	0.7995	0.8886
Random Forest	4096	225739	32309	0	87.67%	1	0.8747	0.9332
AdaBoost	0	258048	0	4096	98.43%	0.9843	1	0.9308
XGBoost	4096	237739	20309	0	92.25%	1	0.9212	0.9590

Fig 20: Results of ML models trained with Unique Normal dataset and tested on Attack dataset

ML Results

Trained using UNOR, Tested on Attack
+VE: 258048, -VE: 4096

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	3133	196758	61290	963	76.25%	0.9951	0.7624	0.8634
Decision Tree	4096	249530	8518	0	96.75%	1	0.9669	0.9832
Random Forest	4096	253795	4252	0	98.37%	1	0.9835	0.9916
AdaBoost	4096	249530	8518	0	96.75%	1	0.9669	0.9668
XGBoost	4096	256456	1592	0	99.39%	1	0.9938	0.9969

Fig 21: Results of ML models trained with UNOR dataset and tested on Attack dataset

ML Results

Trained using Constrained, Tested on Special
+VE: 15000, -VE: 5000

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	5000	6838	8162	0	59.19%	1	0.4558	0.6262
Decision Tree	106	14524	476	4894	73.15%	0.7479	0.9628	0.8439
Random Forest	4982	13402	1598	18	91.92%	0.9986	0.8934	0.9431
AdaBoost	1582	13915	1085	3418	77.48%	0.8028	0.9276	0.8607
XGBoost	4923	14785	215	77	98.54%	0.9948	0.9856	0.9902

Fig 22: Results of ML models trained with Constrained dataset and tested on Special dataset

ML Results

Trained using NCombined, Tested on Special
+VE: 15000, -VE: 5000

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	5000	5	14995	0	25.02%	1	0.0003	0.0006664
Decision Tree	5000	4130	10870	0	45.65%	1	0.2753	0.4317
Random Forest	5000	8351	6649	0	66.75%	1	0.5567	0.7152
AdaBoost	5000	3	14997	0	25.01%	1	0.0002	0.0003999
XGBoost	5000	11565	3435	0	82.82%	1	0.7710	0.8706

Fig 23: Results of ML models trained with NCombined dataset and tested on Special dataset

ML Results

Trained using Unique Normal, Tested on Special
+VE: 15000, -VE: 5000

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	5000	135	14865	0	25.67%	1	0.009	0.01783
Decision Tree	4983	4163	10837	17	45.73%	0.9959	0.2775	0.4340
Random Forest	5000	11809	3191	0	84.04%	1	0.7872	0.8809
AdaBoost	5000	0	15000	0	25.00%	0	0	0
XGBoost	4979	14424	576	21	97.01%	0.9985	0.9616	0.9797

Fig 24: Results of ML models trained with Unique Normal dataset and tested on Special dataset

ML Results

Trained using UNOR, Tested on Special
+VE: 15000, -VE: 5000

Model	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Logistic Regression	4959	7492	7508	41	62.25%	0.9945	0.4994	0.6649
Decision Tree	4346	10943	4057	654	76.44%	0.9436	0.7295	0.8228
Random Forest	4949	14477	523	51	97.13%	0.9964	0.9651	0.9805
AdaBoost	5000	6657	8343	0	58.28%	1	0.4438	0.6147
XGBoost	4966	14483	517	34	97.24%	0.9976	0.9655	0.9813

Fig 25: Results of ML models trained with UNOR dataset and tested on Special dataset

ML Results

Trained using UNOR, Tested on Unconstrained, Attack and Special
ANN: best_params = {"n_layers":4, "first_layer_nodes":256, "last_layer_nodes":4,
"activation_func":"relu" ,"loss_func":"binary_crossentropy", "epochs": 100,
"batch_size": 100}

Test Dataset	TN	TP	FN	FP	Overall statistics			
					Accuracy	Precision	Recall	F1 score
Unconstrained	2	19117	880	1	0.956	0.9999	0.956	0.9775
Attack	4096	255903	2145	0	0.9918	1	0.9917	0.9958
Special	4954	14846	154	46	0.99	0.9908	0.9699	0.9802

Fig 26: Results of ANN ML models trained with UNOR dataset and tested on different testing datasets