A Two-Stage Cyber Attack Detection and Classification System for Smart Grids

Mohammed M. Alani[a], Lara Mauri[b], Ernesto Damiani[b,c]

[a] *Cybersecurity Research Lab, Toronto Metropolitan University, Toronto, Canada*
[b] *Computer Science Department, Università degli Studi di Milano, Milan, Italy*
[c] *Center of Cyber–Physical Systems (C2PS), Khalifa University, Abu Dhabi, United Arab Emirates*

Abstract

As the adoption of Internet of Things (IoT) devices increases rapidly, industrial applications of IoT devices gain further popularity. Some of these applications, such as smart grids, are considered high-risk applications. In the past few years, smart grids became the target of many cyber attacks. In this paper, we present a two-stage system for the detection and classification of cyber attacks based on machine learning. The first stage of the proposed system focuses on detecting attacks efficiently and accurately. The second stage analyzes available data and predicts the specific attack class. The proposed system was tested using the DNP3 intrusion detection dataset, and delivered an F_1 score of 0.9976 at the detection stage, and 0.9883 at the attack type classification stage.

Keywords: attack, intrusion, detection, machine learning, smart grid, dnp3

1. Introduction

Over the last decade, the power-generation sector has undergone a substantial change in the shift from conventional electrical grids to so-called *smart grids* (SGs). This disruptive innovation in the electricity industry is largely due to emerging requirements, like swift population growth and pressing demand for sustainable energy, and to the availability of data-driven intelligence techniques capable of satisfying such requirements. Traditional electric power systems are no longer a practical solution for energy provision and distribution, mainly because of their static operating mode characterized by unidirectional power flows with slow response to outages. In SG, widespread sensor networks supported by communication and information technologies provide a two-way flow of energy and information about the grid status that allows the exchange of measurement data between grid entities [1]. SG allows operators to optimize the power infrastructure in terms of energy consumption, cost, reliability, interoperability, and environmental safeguard. Clearly, hoowever, the merits of SGs necessarily come with an increase in operation complexity.

There is wide consensus that the problem of transitioning traditional power grids into SGs can be tackled using Internet-of-Things (IoT) technology [2]. IoT services can collect, transmit, and process huge amounts of data, with high throughput and low latency. Backed up by Big Data analysis, IoT is a major enabler of SGs, as it provides full control on power quality and dependability [3, 4]. IoT-enabled SGs give power suppliers more efficient and accurate ways to read meters and issue bills. Users enjoy real-time knowledge of (and control over) their energy consumption and even sell directly their surplus to one another.

The National Institute of Standards and Technology (NIST) has recently released the draft of SG framework 4.0 [5], which describes the overall composition of IoT-aided SG systems. The NIST Smart Grid Conceptual Model (SGCM), originally introduced in 2010 [6] and subsequently revised in [7, 8], presents seven different logical domains, namely customer, markets, service provider, operations, generation including distributed energy resources (DER), transmission, and distribution. Each domain describes SG conceptual roles and services, including the interactions among stakeholders needed to perform tasks and achieve system goals. Fig. 1 shows the high-level concepts contained in the latest NIST SGCM, which assist in understanding the various physical and informational interfaces across the SG.

The deployment of advanced cyber-physical systems like SGs is expected to rise significantly over the next few years, especially for use in residential and commercial facilities [3]. However, the connection of massive numbers of devices to communication networks has a huge impact on the security threats' landscape [9]. Increasing wireless connectivity and virtualisation widens

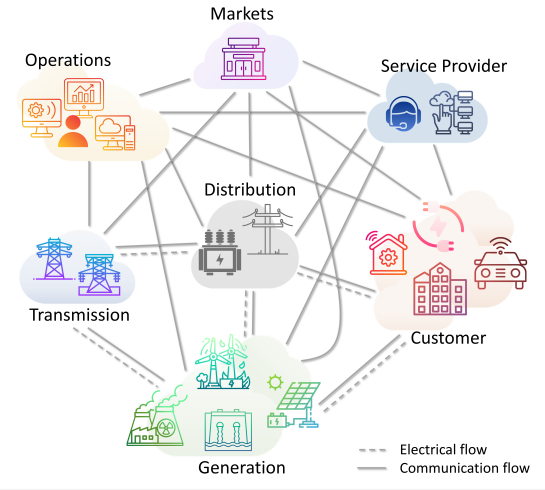

Figure 1: NIST Smart Grid Conceptual Model [7].

the attack surface, opening the doors to cyber-attacks of unprecedented severity [10].

Smart energy systems are the target of a large number of attacks [11]. For instance, injection-type attacks attempt to alter, delete, or insert manipulated data into the network to disrupt the grid operation. Among the cyber-attacks reported in the literature, the false data injection attack (FDIA) is considered one of the most vicious. In FDIAs, the adversary tampers with the meter measurements to interfere with the result of state estimation. Other popular attacks to SGs are jamming and Denial-of-Service (DoS) [12]. In these attacks, the adversary's goal is to keep the channel busy by broadcasting steady or random signals that inhibit transmission and reception by authorized devices. Several surveys discussed the security of SGs and provided classification of prominent cyber-attacks along with their impacts [13, 14]. The literature agrees that developing advanced techniques for fast attack detection is crucial [15, 16]. Most of the available approaches rely on ranging or localization-based techniques or predictive models [17]. Machine Learning-based detection techniques have received widespread attention in recent years, and are considered the most promising ones in terms of effectiveness and scalability [18, 19].

1.1. Research Contribution

This paper presents the following research contributions:

1. A novel technique that cleanly separates the attack detection and attack classification processes to improve detection speed and accuracy. Binary classification is performed first; then, once an attack has been detected, data is passed to the second stage to perform multi-class classification and identify the specific attack type. This arrangement allows for very fast first reaction.
2. A highly efficient and accurate Intrusion Detection System (IDS) focused on detecting attacks to SGs. Our IDS uses only a small number of network flow features (12 features, down from the original 96). The method we used (recursive feature elimination) does not only reduce the number of features fed into the classifier, but also the number of features captured at the data acquisition stage.

1.2. Paper Layout

The remainder of the paper is organized as follows. Section 2 discusses previous relevant works on attack detection in SGs. Section 3 describes the two main stages of our detection and classification mechanism, which quickly and reliably identifies that an attack is ongoing and then determines the specific class of the attack. In Section 4, we experimentally evaluate the effectiveness of our proposed scheme on the DNP3 Intrusion Detection Dataset benchmark, discussing the results in Section 5. Finally, Section 6 provides some concluding remarks and outlines our future research directions.

2. Related Works

In recent years, SG security has received increasing attention due to a widening threat landscape and increasingly frequent attacks. An attack on smart grid systems, for example, could plunge an entire city into darkness. Weak security in smart meters could result in fraud or privacy breaches. Several research lines have been proposed to devise attack detectors and mitigation approaches for SGs. Early research has focused mainly on FDIA identification [20, 21], though some work has addressed different types of attacks that may be carried out on a SG system, such as DoS, DDoS and GPS spoofing attacks [17]. In the following, we present the most relevant previous works, where each paragraph outlines a different case.

According to [22], FDIA detection algorithms can be classified into two categories: *model-based* and *data-driven* methods. In model-based methods, after building a system model, an estimate of the system state is computed with the measurement of the same state in the real system. While model-based methods do not necessarily involve historical data, time factors, such as detection latency, limit their applicability. Data-driven techniques typically do not affect the system and its operation, but

depend on historical data. Also, they require a training process to reduce detection time and increase scalability. These algorithms may involve either learning or conventional time-series mining. Below we present some recent techniques, later summarized in Table 1.

In order to reduce detection latency while ensuring high detection accuracy, many works rely on the Quickest Change Detection (QCD) [23] approach, which detects abrupt changes in the system as quickly as possible. Cumulative sum (CUSUM)-type algorithms are the most commonly used statistical methods for QCD. For instance, Nath et al. [24] developed in 2022 a QCD technique using a normalized Rao-CUSUM test capable of detecting FDIAs while minimizing the worst case detection delay. The algorithm can accurately distinguish FDIAs from sudden system changes. Yet, it is not able to differentiate system faults from FDIAs.

Kurt et al. [25] proposed an online detection algorithm against combined FDIAs and jamming attacks, as well as stealthy attacks against cumulative sum (CUSUM)-based detectors. The authors modelled the SG as a discrete-time linear dynamic system where parameters are initially established together with an attack category. Following a classic approach [26], they used a Kalman filter as the state estimation mechanism to update the model's state based on the measurements. At each step, a state prediction is built based on the state of the previous step. Then, a correction is made to the prediction using the measurements collected in that step.

Wang et al. [27] focused on FDIA localization, treating the localization problem of FDIAs as a multi-label classification task. The authors proposed a Deep-Learning-based Locational Detection (DLLD) architecture to detect the location of FDIAs in real time. Their scheme concatenates a Convolutional Neural Network (CNN) with a standard Bad Data Detector (BDD). The CNN captures the inconsistency and co-occurrence dependency introduced by FDIA, while the BDD estimates real-time measurements' quality and filters out low-quality information.

Shen et al. [28] proposed in 2023 a data-driven localization method also based on a CNN. The authors optimized the CNN using the sparrow search algorithm [29] to select the hyper-parameters. They conducted simulations on the IEEE14-bus and IEEE118-bus test systems, with results exhibiting a localization accuracy of 99.85% and 97.14% in the two systems, respectively, and a false detection rate of only 0.03% in both systems.

In 2021, Siniosoglou et al. [30] presented *MENSA*, an anomaly detection and classification system that combines a classic autoencoder and a Generative Adversarial Network (GAN), dealing respectively with the reconstruction difference and the adversarial error Their model was validated using network traffic and operational data (e.g., time-series electricity measurements) originating from different SG evaluation environments. The authors showed that MENSA is capable of detecting and recognising DNP3 and Modbus/TCP-related cyber attacks and potential operational abnormalities.

Instead of using a standard GAN, Li et al. [31] introduced a new cyber-physical model including an adaptive, window-based GAN. Their scheme integrates a physical model designed to capture ideal measurements with a GAN developed to capture deviations from those measurements. Simulation results show that the proposed technique can accurately recover the state data manipulated by FDIAs.

Kwon et al. [32] presented a behavior-based IDS for IEC 61850 protocol using both statistical analysis of traditional network features and specification-based metrics, while [33] proposed an ML-based IDS targeted at automation networks of substations based on the IEC 60780-5-104 protocol. Similarly, Radoglou et al. [34] proposed anomaly-based IDS called ARIES (smArt gRid Intrusion dEtection System), which combines of three detection layers: (i) network flow-based detection, (ii) packet-based detection, and (iii) operational data-based detection. For each layer, multiple ML/DL methods were adopted, utilising real data originating from power plants.

The authors in [35] proposed a transformer-based intrusion detection model (Transformer-IDM) in which the transformer and feature exaction layers are leveraged to process categorical and numerical features in order to improve the detection performance. They also introduced a hierarchical federated learning intrusion detection system to collaboratively train Transformer-IDM to protect user privacy in the core networks. The obtained intrusion detection model is used by each user to monitor the attacks locally and trigger the alarm in time.

Dou et al. [36] presented in 2022 a hybrid FDIA detection mechanism using temporal correlation to ensure the security of power system operations and control. The proposed mechanism combines ML and Variational Mode Decomposition (VMD) technology. BY VMD, the multiscale spectrum of the SG's states time series is computed and four statistical-based features are extracted from it. The scheme leverages the sequence learning ability of an ensemble classification framework (the OS-Extreme Learning Machine (OSELM)) to identify abnormal states from their prefixes.

Salehpour et al. [37] proposed an attack detection mechanism that can detect cyber-attacks in the early

stages of failure propagation in SG networks. In particular, they utilized a realistic failure propagation (RFP) model [38], which is a graph-based model that uses power and communication networks to study cascading failures. The RFP model serves to generate data for the SVM and NBN algorithms used for fault detection.

The work in [39] presented a supervised data mining technique to detect simulated intrusions on a Modbus network. Neural network and decision trees were employed to classify the traffic generated from the simulated smart factory environment, and the latter were shown to achieve better results.

Inspired by anomaly detection in temperature sensor networks [40], Drayer and Routtenberg [41] proposed a Graph Signal Processing (GSP) technique that calculates the Fourier transform of grid states and filters out the high-frequency elements. Then, FDIAs are detected by comparing the maximum norm of the filtered signal with a threshold. Simulations on the IEEE 14-bus test case showed that the proposed technique facilitates the detection of previously undetectable attacks based on the high-frequency content, the detection precision of which is affected by a proper choice of the threshold. Thus, if the frequency band is too small, attacks might pass undetected.

3. Proposed System

Our system is designed to cleanly separate "attack detection", and "attack type classification" into two separate stages. The reason behind this split is to prioritize the attack detection process over the identification of the attack type, enabling the optimization of the detection process to achieve minimal latency, while not compromising the accuracy of the attack type identification.

Figure 2 shows an overview of the two operational phases of our proposed system.

In the first phase, namely the development phase, the raw dataset is analyzed and processed to produce two datasets; a detection dataset, and a classification dataset. The *detection* dataset is created by converting the attack labels of the original dataset into a binary-labelled dataset with one being the 'attack', covering all types of attacks, and a zero label marking 'benign' samples. The *classification* dataset is created by removing the 'benign' samples, and keeping the attack-type labels for all the 'attack' samples. Both datasets are then preprocessed to ensure that the data is ready for training and testing. This preprocessing includes steps such as removing samples with missing data, and ensuring that all classes are reasonably balanced within the datasets.

After pre-processing, the detection dataset is randomly split to create training and testing subsets. The training subset is used to train a pipeline of classifiers. Then, the best performing classifier is used in selecting the lowest possible number of features while maintaining high accuracy. This process is performed using Recursive Feature Elimination (RFE). This iterative feature selection enables the system to select a small number of highly effective features to improve the detection efficiency. Efficiency improves in two ways; reducing the number of features fed into the classifier, and reducing the number of features that must be extracted at the data acquisition stage during the deployment. The classification dataset is also randomly split to training and testing subsets. The training subset is used to train and test a pipeline of classifiers to select the best performing one in classifying attack types. These best performing classifiers will be pass on to the deployment phase.

In the deployment phase, network traffic is captured by a packet-capturing unit. This unit utilizes lightweight tools such as "tcpdump", to keep the overhead to a minimum. The captured packets are then processed by a feature extraction unit designed to extract network flow features from raw network packets. This unit utilizes two tools named "CICFlowMeter" and "DNP3 parser", as described in [42]. The extracted features include network flow features such as source and destination port numbers, number of packet in the network flow, data rate per second, in addition to DNP3 specific features. These features are carefully selected to help the classifiers maintain high accuracy while minimizing the data captured and processed. A list of all features can be found in [43]. Once the features are extracted, they are passed to the pre-trained attack *detection* classifier, that resembles stage 1 in the system. This classifier produces a prediction whether this flow should be considered an attack or not. If the flow is detected as an attack, the extracted features are passed over to the second stage where the second pre-trained classifier identifies the specific type of attack. Identifying the specific type of attack supports decision making on countermeasures and mitigation actions. In addition, identification of the specific attack type can help in forensic analysis after the detection.

Table 1: Summary of previous work reviewed

Reference	Victim System	Attack Type	Dataset Generator	Solution Method
[24]	SCADA system, Phasor measurement units, and Intelligent electronic devices	FDI-Power grid state transitions and worst case detection delays	13-bus system	Quickest intrusion detection algorithm and Dynamic state estimation algorithm
[25]	Smart grid	FDI/Jamming-Power measurements	IEEE 14-bus system	Online CUSUM-based detection and estimation algorithm
[27]	Power system state estimator	FDI-Power buses	IEEE 14- and 118-bus systems	Deep-learning-based locational detection algorithm
[28]	Smart grid	FDI-Power buses and lines	IEEE 14- and 118-bus systems	Sparrow search algorithm and CNN classification model
[30]	Smart grid	Modbus/TCP- and DNP3-related attacks and anomalies	Modbus/TCP, DNP3 and operational data	Autoencoder-GAN-based algorithm
[31]	Power system state estimator	FDI-Power measurements and sensors	IEEE 30- and 118-bus systems	Online GAN-based cyber-physical model
[32]	Smart grid	Digital substation LAN-related attacks	IEC 61850 based network traffic	Behavior-based intrusion detection system
[33]	Electrical substations	Passive eavesdropping, Reprogramming, and DoS attacks	IEC-60870-5-104-based network traffic	ML-based anomaly detection algorithm
[34]	Smart grid	Modbus/TCP-related and operational data attacks	CSE-CIC-IDS2018	Anomaly-based intrusion detection system
[35]	Smart grid	DoS, Probing scanning, Remote to local, and User to root attacks	NSL-KDD	Transformer-based intrusion detection model
[36]	Power system state estimator	FDI-Power buses and sensors	IEEE 14-bus system	Online sequential extreme learning machine and Variational mode decomposition
[37]	Smart grid	DoS attacks and FDI-Power measurements	IEEE 118-bus system	Supervised early attack detection algorithm based on RFP model
[39]	Modbus network	Modbus-related attacks	Malicious traffic injection	Data mining-based intrusion detection
[41]	AC Power system	FDI-Power measurements	IEEE 14-bus system	Graph signal processing-based detection algorithm
Proposed work	Smart grid	DNP3-related attacks	DNP3	A two-stage attack detection and classification using RF and XGB

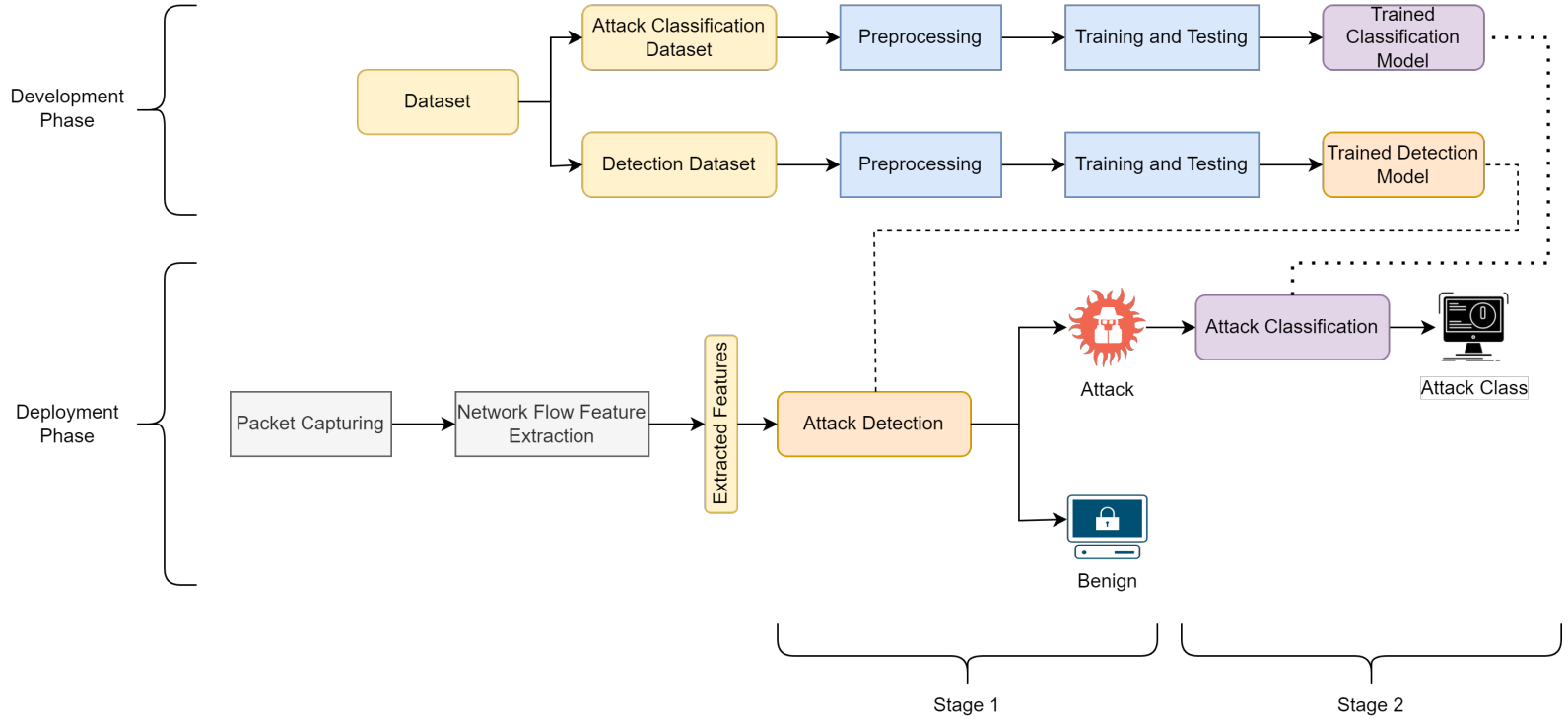

Figure 2: Proposed System Overview

4. Experiments and Results

In this section, we discuss our experiment's design, implementation, and results.

4.1. Experiment Design

1. The first step in our experiment is to create two copies of the original DNP3 intrusion detection dataset; the first one is used for attack detection, and the second is used for identifying the attack's class. The attack detection dataset was created by relabelling all attack types into an 'attack' label, and labeling all normal traffic samples to 'benign'. The second dataset was created by removing all normal traffic and keeping only traffic labeled with one of the attack labels.
2. Preprocessing both datasets to ensure that they do not suffer from significant imbalance or missing data.
3. Creating the pipeline for selecting binary ML classifiers using the attack detection dataset, and testing it to select the best performing classifier to be used in the following feature selection step.
4. Selecting the lowest possible number of features while maintaining high accuracy. We use RFE method to iteratively eliminate the feature with the lowest feature importance according to Algorithm 1, producing a dataset with a reduced number of features.
5. Creating a second pipeline of classifiers to test the outcome of the feature selection process and ensure that it does not cause significant performance degradation. At the end of this step, the first stage (the attack detection classifier) is complete.
6. For the attack classification stage, a new pipeline of multi-class classifiers is created, trained, and tested using the second dataset.
7. The best performing classifier undergoes hyperparameter optimization to improve its performance.
8. Both of the selected classifiers undergo 10-fold cross-validation to ensure that they are capable of generalizing well beyond their training datasets.

4.2. Experimentation Environment

All experiements were conducted on a computer with the following specifications:

- Processor: AMD Ryzen 5 3600 4.2GHz
- RAM: 128GB

Algorithm 1: Recursive Feature-Elimination Using Feature Importance

Input: Dataset with m features
Output: Dataset with n features

$Array \leftarrow Dataset$
$model = MLClassifier$
$TargetFeatures = n$
while $Features(Dataset) > TargetFeatures$ **do**
  train $model$ with $Array$
  $importance = FeatureImportance(model)$
  i = index of feature with lowest importance
  $Array.DeleteFeature(i)$
end
Store $Array \rightarrow Dataset$

- OS: Windows 10 Professional
- Python v3.10
- SciKit Learn v1.1.3
- XGBoost v1.5.0
- Numpy v1.23.4
- Pandas v1.5.2

4.3. DNP3 Intrusion Detection Dataset

In this paper, we rely on the DNP3 Intrusion Detection Dataset [43] to evaluate the performance of our two-stage attack detection and classification mechanism. This dataset was curated by Radoglou-Grammatikis et al. [44], following the methodological frameworks proposed by Gharib et al. [45] and Dadkhah et al. [46].

To generate the DNP3 dataset, a network topology consisting of *(i)* eight industrial entities, *(ii)* one Human Machine Interfaces (HMI), and *(iii)* three cyber-attackers was employed to capture and represent attacks conducted in TCP/IP flows and DNP3-specific flows, as shown in Fig. 3. In the testbed utilized for the implementation, the industrial entities play the role of the DNP3 outstations/slaves (Remote Terminal Units and Intelligent Electron Devices), while the further workstation played the role of the master (Master Terminal Unit).

The following nine DNP3 cyber-attacks, involving DNP3 unauthorized commands and Denial of Service (DoS), were implemented using popular penetration testing tools like Nmap and Scapy:

- DNP3 Disable Unsolicited Messages Attack

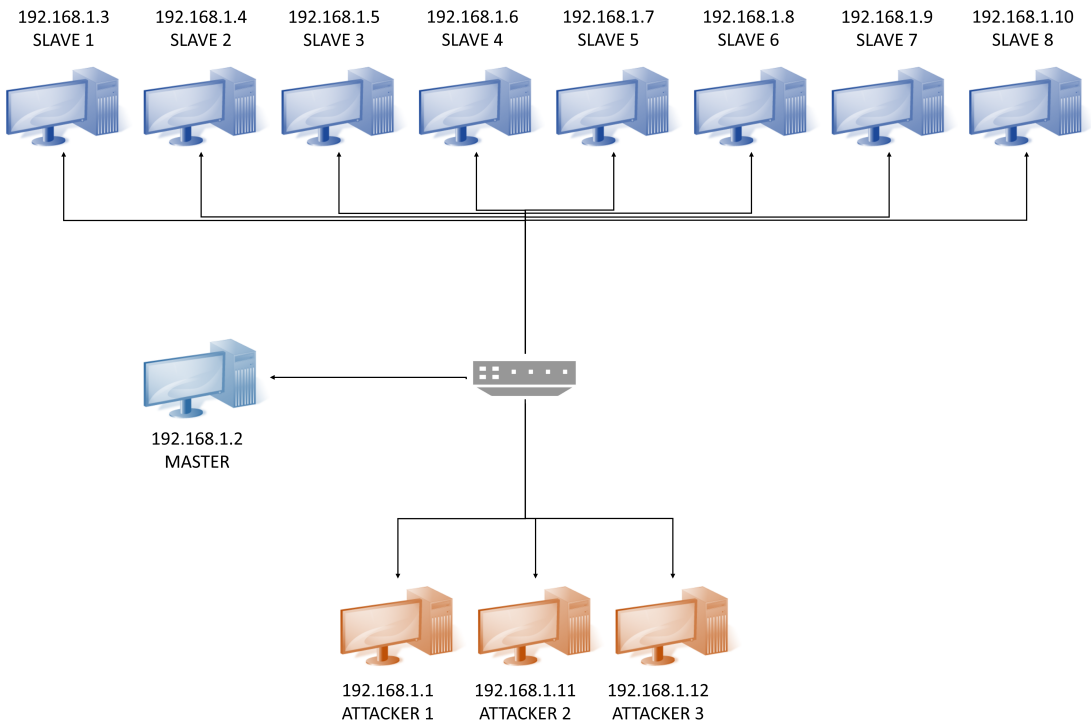

Figure 3: Testbed for the DNP3 Intrusion Detection Dataset generation [43]

- DNP3 Cold Restart Message Attack
- DNP3 Warm Restart Message Attack
- DNP3 Enumerate Attack
- DNP3 Info Attack
- DNP3 Initialisation Attack
- Man In The Middle (MITM)-DoS Attack
- DNP3 Replay Attack
- DNP3 Stop Application Attack

The DNP3 and TCP/IP network flows generated by each node during the attack execution were produced by using a custom DNP3 Python Parser and the CICFlowMeter, respectively [42]. The resulting dataset, consisting of the above flows and related statistics, was labelled based on the previously listed DNP3 attack types. We decided to remove the MITM-DoS attack from the extracted files, as it does not contain the same features extracted from the other attacks, apparently due to an extraction error. The dataset extracted from the remaining captured network packets included 40,420 samples. Each sample, carries 101 features extracted from the network flows, labelled into 8 different attacks.

4.4. Preprocessing and Sub-Datasets Creation

Upon examining the dataset, we made the following observations:

- The dataset contains multiple host-specific features such as source IP address, and destination IP address.
- The dataset includes features with string values such as `firstPacketDIR` that show whether the flow started at the master node or the slave node.

Host-related data are a major cause of overfitting [47]. We removed all host-specific features from the dataset to to help our trained classifiers to generalize beyond this dataset. In addition, we removed other irrelevant features such as the flow number, flow ID, and date of the flow. The next step in pre-processing was to numerically encode all string-based features. This created a dataset with 96 features, and 40,420 samples.

As our proposed design utilizes two classifiers in two separate stages, we created two datasets from the preprocessed one:

1. The first dataset includes two labels only; malicious, and benign. This dataset is used in stage

1 for the detection process. To create it, we removed all attack-specific labels and replaced them with "malicious". As this step was done, we noticed that the normal class contained 14,380 samples, while the malicious class contained 26,040 samples. To address this imbalance, we performed random oversampling of the minority class to get to a stage 1 dataset with 96 features, and 52,80 samples (26,040 normal, and 26,040 malicious). This moderate rate of random oversampling only marginally increases the likelihood of overfitting, which is already taken care of by the feature preprocessing discussed above.

2. The second dataset includes eight attack labels to be used in identifying the specific attack type. Based on that, we removed the "benign" samples from the second dataset. Upon the removal of the normal samples, we noticed a limited imbalance between the different attack types. Hence, we performed again random over sampling for the classes with lower number of samples. This resulted in a second dataset of 46,080 samples (5,760 samples in each class).

4.5. Stage-One: Attack Detection

As stated in Section 4.1, in the first step in stage 1, we created a pipeline of binary classifiers for the attack detection stage. This pipeline includes the following classifiers:

- Random Forest (RF)
- Logistic Regression (LR)
- Decision Tree (DT)
- Gaussian Naive-Bayes (GNB)
- Extreme Gradient Boosting (XGB)

The purpose of this stage is to find the best performing classifier to use it later in the feature selection process. We performed a stratified random split to the stage 1 dataset to select 75% of the samples for training, and 25% of the samples for testing. This technique consists of forcing data distribution to be the same across different dataset splits. This way, classifiers are trained on the same population where they are evaluated, achieving better predictions. Table 2 shows the testing results of the initial stage-1 classifiers pipeline.

As shown in the table, the RF classifier outperformed the others in terms of accuracy and F_1 score. Based on these results, we selected RF classifier for the feature selection process (RFE) shown in Algorithm 1. RFE identifies the feature with the lowest importance and eliminates it from the dataset, and then repeats the training and testing. Then, the algorithm keeps repeating the above-mentioned steps until a the number of features is reached beyond which the performance of the classifier drops significantly. Figure 4 shows the change in F_1 score of the classifier with the process of feature elimination.

Table 2: Initial testing results for attack detection using 96 features

Metric	Accuracy	Precision	Recall	F_1 Score
RF	0.997696	0.997698	0.997696	0.997696
LR	0.946390	0.946803	0.946390	0.946378
DT	0.977158	0.977163	0.977158	0.977158
GNB	0.629339	0.760120	0.629339	0.576052
XGB	0.988464	0.988464	0.988464	0.988464

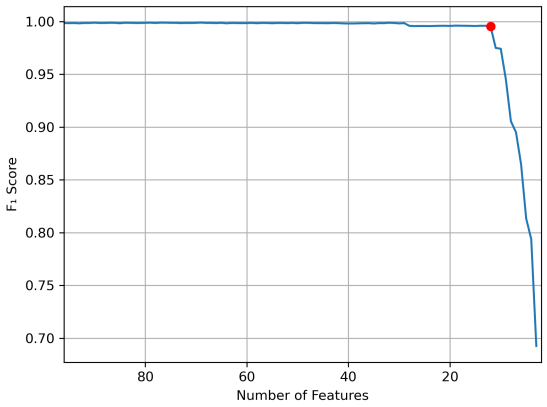

Figure 4: Change in F_1 score with feature reduction

As shown in the figure, reducing the number of features below 12 results in significant drop in the F_1 value. Hence, the number of features we selected based on RFE algorithm was 12.

In the next step, the reduced stage 1 dataset was used to train and test the classifiers pipeline to ensure that this significant reduction in features did not impact the classifier's performance. Table 3 shows the testing results obtained using the 12-feature stage 1 dataset.

As shown in the table, RF, DT, GNB, and XGB do not show any significant drop in performance metrics. However, LR witnessed a significant drop in accuracy and F_1 score. Nvertheless, the feature selection stage was successful in reducing the number of features from 96 to 12 without a significant impact on the best per-

Table 3: Detection results after feature selection

Metric	Accuracy	Precision	Recall	F_1 Score
RF	0.995008	0.995009	0.995008	0.995008
LR	0.625499	0.751297	0.625499	0.571926
DT	0.973779	0.973785	0.973779	0.973779
GNB	0.628034	0.767301	0.628034	0.572328
XGB	0.985238	0.985240	0.985238	0.985238

Table 4: Attack classification testing results

Metric	Accuracy	Precision	Recall	F_1 Score
RF	0.975955	0.975965	0.975955	0.975955
LR	0.538889	0.551916	0.538889	0.517928
DT	0.977240	0.977273	0.977240	0.977239
GNB	0.568663	0.550899	0.568663	0.505081
XGB	0.985503	0.985523	0.985503	0.985503

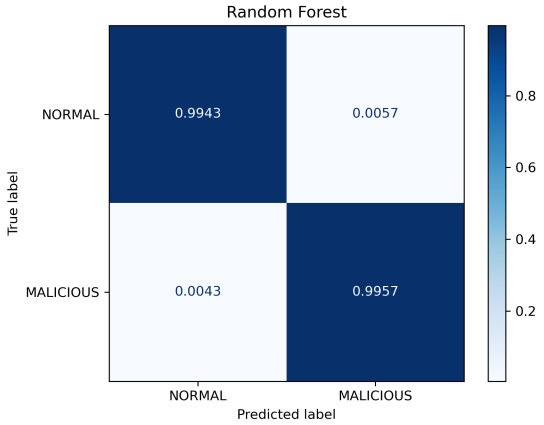

Figure 5: Attack detection stage confusion matrix plot with 12 features

forming classifier, namely RF. Figure 5 shows the confusion matrix plot for the RF classifier using 12 features.

Our system produced a superior false-positive detection rate of 0.57%, and a false-negative rate of 0.43% only.

Also, the testing time per instance dropped from 13.67μs using 96 features to 6.12μs using 12 features. It is notable that the F_1 score of the system had a minor drop from 0.9976 to 0.9950 (merely 0.2%), while the instance processing time dropped by 55%.

4.6. Stage-Two: Attack Classification

As described in Section 4.1, the second stage starts by creating a new multi-class classifiers pipeline to find the best performing one. The stage 2 dataset was randomly split into 75% training subset and 25% testing subset, with stratification. Table 4 shows the attack classification testing results for the trained pipeline.

As shown in the table, the best performing multi-class classifier is XGB with F_1 score of 0.9855. As per our experiment design, the next stage is to perform hyperparameter optimization to improve the performance of the classifier.

The optimization step resulted in selecting the following hyperparameters:

Table 5: Attack classification results after hyperparameter optimization of XGB classifier

Metric	Value
Accuracy	0.988281
Precision	0.988281
Recall	0.988281
F_1 **Score**	0.988281

- max_depth = 15
- learning_rate = 0.2
- subsample = 0.799999
- colsample_bytree = 0.799999
- solsample_bylevel = 0.5
- n_estimators = 100

The results obtained by using the selected hyperparameters are shown in Table 5. As shown in the table, the F_1 score slightly improved from 0.9855 to 0.9882.

Figure 6 shows the confusion matrix plot of the optimized XGB classifier used in the second stage of our system.

As shown in the figure, the trained classifier performs perfectly in identifying six out of the eight attack classes. The two classes in which the performance is less than perfect are warm restart, and info attacks. The classifier mis-classified about 5% of the samples from the warm restart to the info attack class, and vice-versa.

4.7. Cross-Validation

To verify the reliability of the experiment results, both of our classifiers (stage 1 and stage 2) were passed through a 10-fold validation step. Within this step, the dataset of the classifier is split into 10 folds. Each fold is used for testing once, while the other nine are used for training. The classifier undergoes this cycle ten times. If the average of the resulting performance metrics is close to the results obtained previously, the results are considered reliable and the classifier can generalize well beyond its training dataset.

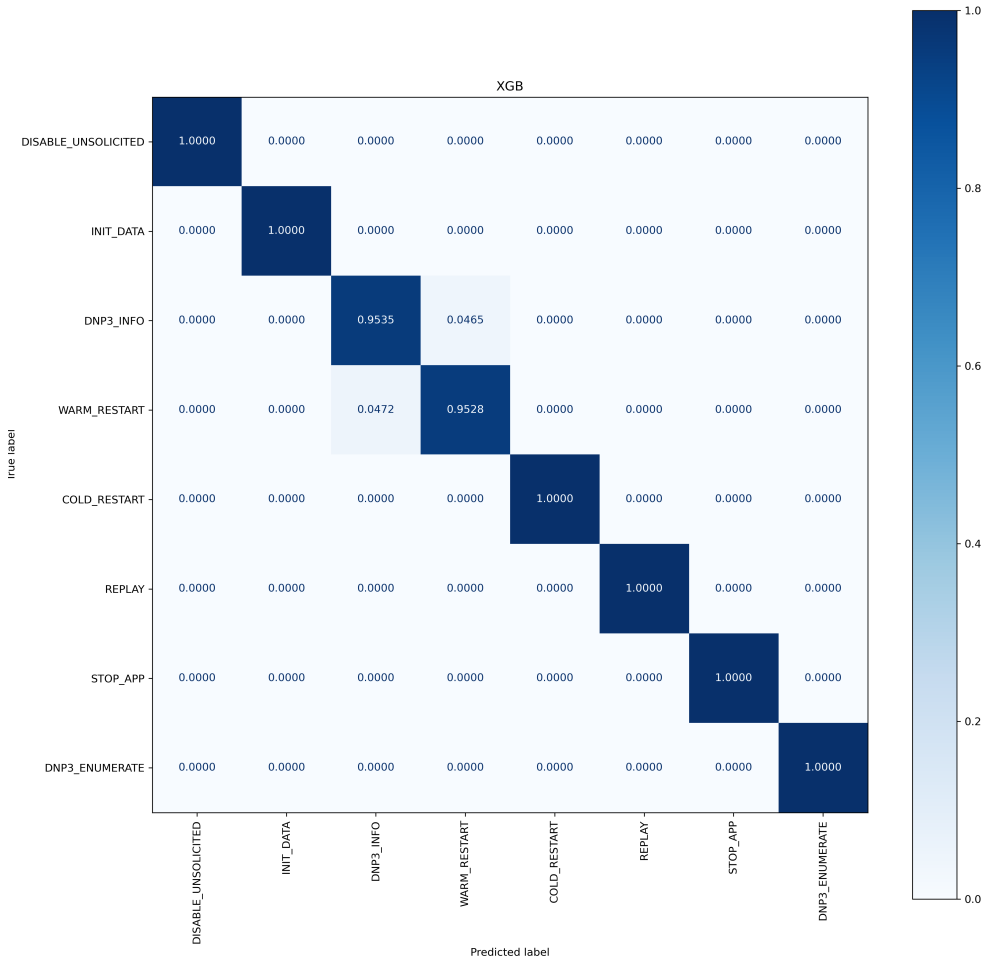

Figure 6: Attack classification stage confusion matrix plot

Table 6 shows the mean and standard deviation of the performance metrics for the RF classifier of stage 1, and for the XGB classifier of stage 2.

As shown the table, the results obtained in the 10-fold cross-validation for both stage 1 and stage 2 are consistent with the results obtained earlier. In addition, the low standard deviation confirms that our system can generalize well beyond its training dataset.

5. Discussions

Poor generalization has been called the Achilles heel of supervised systems [48], whose performance can become disappointing in production when features' distributions change [49]. As explained in Section 4, our testing results show that the results we obtained are robust, and that our system is capable of generalizing well beyond its training dataset. Table 7 shows a comparison of the performance metrics of our system with that of relevant related works.

As shown in the table, the detection metrics of our proposed system exceed those of [33], significantly. While our metrics are comparable to [39], the dataset utilized in that paper suffers from significant imbalance. The normal traffic samples captured was 92% of the total accuracy. This imbalance causes a wildly inaccurate measure of its accuracy, where it achieves a nominal 0.92 accuracy just by classifying all traffic as normal, and not detecting any attack.

When compared to [32], our system slightly outperforms it. However, the number of samples used in testing in that paper is 288 only, because the system proposed in that paper is not a full ML-based IDS. This low number impacts the reliability of the results shown in [32].

As we compare our proposed system with [34], we notice that the dataset used in that paper is a general IDS dataset, and does not include IoT, IIoT, nor smart grid traffic. While the approach seems promising, its validation does not take into account SG-specific traffic data such as low- asymmetric flows, with large amount of data going from sensors to control systems, and small amount of data in the reverse direction. We argue that IDS systems trained with non-SG traffic only may not be capable of thwarting threats that are unique to the smart grid.

We remark that could not compare the detection time with related works because we could not find this parameter in the previous research we reviewed.

6. Conclusions and Future Work

Threats against SGs are becoming increasingly common, especially with the adoption of IoT technology. Despite IoT success in monitoring and controlling the operation of energy system, its complexity widens the attack surface and makes smart devices vulnerable to a multitude of cyber-attacks. Designing effective detection techniques is of paramount importance to ensure the security and resiliency of SGs.

In this paper, we designed a ML-based attack detection and classification system based on a two-step procedure that creates two distinct pipelines of binary and multi-class classifiers. We performed extensive experimental evaluation on the DNP3 intrusion detection dataset. Results have shown that our proposed scheme can quickly spot attacks and accurately identify their type. Also the system generalizes well and promises to be robust with respect to in-production variations of traffic. In our future work, we plan to explore the following areas:

- Using deep packet analysis to detect packet-based attack, such as reconnaissance attacks.
- Using deep neural networks and compare their results to classical ML.
- Explore incorporating attacks on other protocols to generalize our solution to multi-protocol environments.

References

[1] S. R. Salkuti, P. Ray, S. Pagidipala, Overview of Next Generation Smart Grids, Springer Nature Singapore, Singapore, 2022, pp. 1–28. doi:10.1007/978-981-16-7794-6_1.

[2] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, O. Rana, The internet of things, fog and cloud continuum: Integration and challenges, Internet of Things 3 (2018) 134–155.

[3] A. Goudarzi, F. Ghayoor, M. Waseem, S. Fahad, I. Traore, A survey on iot-enabled smart grids: Emerging, applications, challenges, and outlook, Energies 15 (19) (2022) 6984. doi:10.3390/en15196984.

[4] A. Ghasempour, Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges, Inventions 4 (1) (2019) 22. doi:10.3390/inventions4010022.

[5] A. Gopstein, C. Nguyen, C. O'Fallon, N. Hastings, D. A. Wollman, Nist framework and roadmap for smart grid interoperability standards, release 4.0 (2021-02-18 00:02:00 2021). doi:https://doi.org/10.6028/NIST.SP.1108r4.

[6] G. Arnold, D. Wollman, G. FitzPatrick, D. Prochaska, D. Holmberg, D. Su, A. H. Jr., N. Golmie, T. Brewer, M. Bello, P. Boynton, Nist framework and roadmap for smart grid interoperability standards, release 1.0 (2010-01-10 00:01:00 2010). doi:https://doi.org/10.6028/NIST.sp.1108.

Table 6: Results of 10-fold cross-validation

Classifier		Accuracy	Precision	Recall	F_1 Score
Stage 1	Mean	0.996102	0.997462	0.994736	0.996097
	StDev	0.000589	0.000943	0 .001243	0.000593
Stage 2	Mean	0.988711	0.988710	0.988710	0.988710
	StDev	0.000519	0.000733	0.001208	0.000511

Table 7: Comparison of the proposed system to related works

Related work	Balanced	Attacks	Classifier	Detection	Classification	Accuracy	F_1 Score
Hodo et al. [33]	✓	3	J48	✓	✗	0.9169	0.9200
Li et al. [39]	✗	4	J48	✗	✓	0.9983	-
Kwon et al. [32]	✗	6	Hybrid	✓	✗	0.9890	-
Radoglou et al. [34]	✗	7	RF	✓	✗	0.9900	0.9700
Proposed system	✓	8	RF	✓		0.9950	0.9950
		96	XGB		✓	0.9882	0.9882

[7] G. Arnold, G. FitzPatrick, D. Wollman, T. Nelson, P. Boynton, G. Koepke, A. H. Jr., C. Nguyen, J. Mazer, D. Prochaska, M. Swanson, T. Brewer, V. Pillitteri, D. Su, N. Golmie, E. Simmon, A. Eustis, D. Holmberg, S. Bushby, M. Janezic, A. Jillavenkatesa, Nist framework and roadmap for smart grid interoperability standards, release 2.0 (2012-02-16 00:02:00 2012). doi:https://doi.org/10.6028/NIST.sp.1108r2.

[8] C. Greer, D. Wollman, D. Prochaska, P. Boynton, J. Mazer, C. Nguyen, G. FitzPatrick, T. Nelson, G. Koepke, A. H. Jr., V. Pillitteri, T. Brewer, N. Golmie, D. Su, A. Eustis, D. Holmberg, S. Bushby, Nist framework and roadmap for smart grid interoperability standards, release 3.0 (2014-10-01 00:10:00 2014). doi:https://doi.org/10.6028/NIST.SP.1108r3.

[9] L. Baresi, D. F. Mendonça, M. Garriga, S. Guinea, G. Quattrocchi, A unified model for the mobile-edge-cloud continuum, ACM Transactions on Internet Technology (TOIT) 19 (2) (2019) 1–21.

[10] A. Djenna, S. Harous, D. E. Saidouni, Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure, Applied Sciences 11 (10) (2021) 4580. doi:10.3390/app11104580.

[11] T. Talaei Khoei, H. Ould Slimane, N. Kaabouch, A Comprehensive Survey on the Cyber-Security of Smart Grids: Cyber-Attacks, Detection, Countermeasure Techniques, and Future Directions, arXiv e-prints (2022) arXiv:2207.07738arXiv:2207.07738, doi:10.48550/arXiv.2207.07738.

[12] P. Srikantha, D. Kundur, Denial of service attacks and mitigation for stability in cyber-enabled power grid, in: 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), IEEE, 2015, pp. 1–5.

[13] P. Haji Mirzaee, M. Shojafar, H. Cruickshank, R. Tafazolli, Smart grid security and privacy: From conventional to machine learning issues (threats and countermeasures), IEEE Access 10 (2022) 52922–52954. doi:10.1109/ACCESS.2022.3174259.

[14] J. Ding, A. Qammar, Z. Zhang, A. Karim, H. Ning, Cyber threats to smart grids: Review, taxonomy, potential solutions, and future directions, Energies 15 (18) (2022) 6799. doi:10.3390/en15186799.
URL http://dx.doi.org/10.3390/en15186799

[15] O. Tushkanova, D. Levshun, A. Branitskiy, E. Fedorchenko, E. Novikova, I. Kotenko, Detection of cyberattacks and anomalies in cyber-physical systems: Approaches, data sources, evaluation, Algorithms 16 (2) (2023) 85. doi:10.3390/a16020085.
URL http://dx.doi.org/10.3390/a16020085

[16] M. Elnour, N. Meskin, K. Khan, R. Jain, Application of data-driven attack detection framework for secure operation in smart buildings, Sustainable Cities and Society 69 (2021) 102816. doi:https://doi.org/10.1016/j.scs.2021.102816.

[17] U. Inayat, M. F. Zia, S. Mahmood, T. Berghout, M. Benbouzid, Cybersecurity enhancement of smart grid: Attacks, methods, and prospects, Electronics 11 (23) (2022) 3854. doi:10.3390/electronics11233854.
URL http://dx.doi.org/10.3390/electronics11233854

[18] L. Cui, Y. Qu, L. Gao, G. Xie, S. Yu, Detecting false data attacks using machine learning techniques in smart grid: A survey, Journal of Network and Computer Applications 170 (2020) 102808. doi:https://doi.org/10.1016/j.jnca.2020.102808.

[19] Z. Zhang, H. Ning, F. Shi, F. Farha, Y. Xu, J. Xu, F. Zhang, K.-K. R. Choo, Artificial intelligence in cyber security: research advances, challenges, and opportunities, Artificial Intelligence Review (2022) 1–25.

[20] M. Mohammadpourfard, A. Khalili, I. Genc, C. Konstantinou, Cyber-resilient smart cities: Detection of malicious attacks in smart grids, Sustainable Cities and Society 75 (2021) 103116. doi:https://doi.org/10.1016/j.scs.2021.103116.

[21] A. Sayghe, Y. Hu, I. Zografopoulos, X. Liu, R. G. Dutta, Y. Jin, C. Konstantinou, A survey of machine learning methods for detecting false data injection attacks in power systems, CoRR abs/2008.06926 (2020). arXiv:2008.06926.
URL https://arxiv.org/abs/2008.06926

[22] A. S. Musleh, G. Chen, Z. Y. Dong, A survey on the detection algorithms for false data injection attacks in smart grids, IEEE Transactions on Smart Grid 11 (3) (2020) 2218–2234. doi:10.1109/TSG.2019.2949998.

[23] H. Poor, O. Hadjiliadis, Quickest detection, Vol. 9780521621045, Cambridge University Press, United Kingdom, 2008, publisher Copyright: © Cambridge University Press 2009. doi:10.1017/CBO9780511754678.

[24] S. Nath, I. Akingeneye, J. Wu, Z. Han, Quickest detection of false data injection attacks in smart grid with dynamic models, IEEE Journal of Emerging and Selected Topics in Power Electronics 10 (1) (2022) 1292–1302. doi:10.1109/JESTPE.2019.2936587.

[25] M. N. Kurt, Y. Yılmaz, X. Wang, Real-time detection of hybrid and stealthy cyber-attacks in smart grid, IEEE Transactions on Information Forensics and Security 14 (2) (2019) 498–513. doi:10.1109/TIFS.2018.2854745.

[26] J. Zhang, G. Welch, G. Bishop, Z. Huang, A two-stage kalman filter approach for robust and real-time power system state estimation, IEEE Transactions on Sustainable Energy 5 (2) (2013) 629–636.

[27] S. Wang, S. Bi, Y.-J. A. Zhang, Locational detection of the false data injection attack in a smart grid: A multilabel classification approach, IEEE Internet of Things Journal 7 (9) (2020) 8218–8227. doi:10.1109/JIOT.2020.2983911.

[28] K. Shen, W. Yan, H. Ni, J. Chu, Localization of false data injection attack in smart grids based on ssa-cnn, Information 14 (3) (2023) 180. doi:10.3390/info14030180.
URL http://dx.doi.org/10.3390/info14030180

[29] F. S. Gharehchopogh, M. Namazi, L. Ebrahimi, B. Abdollahzadeh, Advances in sparrow search algorithm: a comprehensive survey, Archives of Computational Methods in Engineering 30 (1) (2023) 427–455.

[30] I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras, P. Sarigiannidis, A unified deep learning anomaly detection and classification approach for smart grid environments, IEEE Transactions on Network and Service Management 18 (2) (2021) 1137–1151. doi:10.1109/TNSM.2021.3078381.

[31] Y. Li, Y. Wang, S. Hu, Online generative adversary network based measurement recovery in false data injection attacks: A cyber-physical approach, IEEE Transactions on Industrial Informatics 16 (3) (2020) 2031–2043. doi:10.1109/TII.2019.2921106.

[32] Y. Kwon, H. K. Kim, Y. H. Lim, J. I. Lim, A behavior-based intrusion detection technique for smart grid infrastructure, in: 2015 IEEE Eindhoven PowerTech, IEEE, 2015, pp. 1–6.

[33] E. Hodo, S. Grebeniuk, H. Ruotsalainen, P. Tavolato, Anomaly detection for simulated iec-60870-5-104 trafiic, in: Proceedings of the 12th international conference on availability, reliability and security, 2017, pp. 1–7.

[34] P. Radoglou Grammatikis, P. Sarigiannidis, G. Efstathopoulos, E. Panaousis, Aries: A novel multivariate intrusion detection system for smart grid, Sensors 20 (18) (2020) 5305.

[35] X. Sun, Z. Tang, M. Du, C. Deng, W. Lin, J. Chen, Q. Qi, H. Zheng, A hierarchical federated learning-based intrusion detection system for 5g smart grids, Electronics 11 (16) (2022) 2627. doi:10.3390/electronics11162627.
URL http://dx.doi.org/10.3390/electronics11162627

[36] C. Dou, D. Wu, D. Yue, B. Jin, S. Xu, A hybrid method for false data injection attack detection in smart grid based on variational mode decomposition and os-elm, CSEE Journal of Power and Energy Systems 8 (6) (2022) 1697–1707. doi:10.17775/CSEEJPES.2019.00670.

[37] A. Salehpour, I. Al-Anbagi, K.-C. Yow, X. Cheng, A supervised early attack detection mechanism for smart grid networks, in: 2023 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2023, pp. 1–5. doi:10.1109/ISGT51731.2023.10066351.

[38] A. Salehpour, I. Al-Anbagi, K.-C. Yow, X. Cheng, Modeling cascading failures in coupled smart grid networks, IEEE Access 10 (2022) 81054–81070. doi:10.1109/ACCESS.2022.3194989.

[39] S.-C. Li, Y. Huang, B.-C. Tai, C.-T. Lin, Using data mining methods to detect simulated intrusions on a modbus network, in: 2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2), IEEE, 2017, pp. 143–148.

[40] A. Sandryhaila, J. M. F. Moura, Discrete signal processing on graphs: Frequency analysis, IEEE Transactions on Signal Processing 62 (12) (2014) 3042–3054. doi:10.1109/TSP.2014.2321121.

[41] E. Drayer, T. Routtenberg, Detection of false data injection attacks in smart grids based on graph signal processing, IEEE Systems Journal 14 (2) (2020) 1886–1896. doi:10.1109/JSYST.2019.2927469.

[42] V. Kelli, P. Radoglou-Grammatikis, A. Sesis, T. Lagkas, E. Fountoukidis, E. Kafetzakis, I. Giannoulakis, P. Sarigiannidis, Attacking and defending dnp3 ics/scada systems, in: 2022 18th International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, 2022, pp. 183–190.

[43] P. Radoglou-Grammatikis, V. Kelli, T. Lagkas, V. Argyriou, P. Sarigiannidis, Dnp3 intrusion detection dataset, IEEE Dataport (2022). doi:10.21227/s7h0-b081.
URL https://dx.doi.org/10.21227/s7h0-b081

[44] P. Radoglou-Grammatikis, P. Sarigiannidis, G. Efstathopoulos, P.-A. Karypidis, A. Sarigiannidis, Diderot: An intrusion detection and prevention system for dnp3-based scada systems, in: Proceedings of the 15th International Conference on Availability, Reliability and Security, ARES '20, Association for Computing Machinery, New York, NY, USA, 2020. doi:10.1145/3407023.3409314.
URL https://doi.org/10.1145/3407023.3409314

[45] A. Gharib, I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani, An evaluation framework for intrusion detection dataset, in: 2016 International Conference on Information Science and Security (ICISS), 2016, pp. 1–6. doi:10.1109/ICISSEC.2016.7885840.

[46] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, A. A. Ghorbani, Towards the development of a realistic multidimensional iot profiling dataset, in: 2022 19th Annual International Conference on Privacy, Security and Trust (PST), 2022, pp. 1–11. doi:10.1109/PST55820.2022.9851966.

[47] J. Liu, M. Simsek, B. Kantarci, M. Bagheri, P. Djukic, Collaborative feature maps of networks and hosts for ai-driven intrusion detection, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 2662–2667.

[48] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, F. De Turck, Towards model generalization for intrusion detection: Unsupervised machine learning techniques, Journal of Network and Systems Management 30 (2022) 1–25.

[49] L. Mauri, E. Damiani, Estimating degradation of machine learning data assets, ACM Journal of Data and Information Quality (JDIQ) 14 (2) (2021) 1–15.