Cyber-Physical Defense in Smart Distribution Networks

Leen Al Homoud, Rinith Reghunath, Safin Bayes, Aaqib Peerzada, Katherine Davis, *Senior Member, IEEE,* and Robert S. Balog, *Senior Member, IEEE*

***Abstract*— The existing electric grid is transitioning to a smart grid with increased penetration of distributed energy resources (DERs), such as photovoltaic (PV) units, battery storage units, electric vehicles (EV), and EV chargers. DERs facilitate the increase in renewable energy generation, which leads to a more sustainable, efficient, and reliable grid paradigm. However, with the rise of communication exchanges and data flow due to DERs, cybersecurity vulnerabilities arise. Much of the literature has focused strictly on mitigating data attacks resulting in non-technical losses, false state estimation, and inaccurate load forecasting. However, the grid paradigm's cyber-physical security also needs to be considered to ensure that no grid operations take place that impact the physics of the system. Our project achieved that by developing a Machine Learning (ML) algorithm that will detect anomalies in the commands issued to the distribution network's assets. The algorithm was trained using data from a base case obtained from the simulation of the IEEE 34 distribution network. It was tested and improved by adding modifications to the base case. We successfully developed a local anomaly detection algorithm for a photovoltaic system and two voltage regulators, achieving F1-scores of 0.5141, 0.8173, and 0.8982, respectively. All three algorithms achieved low values of false negatives, which is promising as false negatives have a much higher cost since missing one anomaly can result in disastrous effects on the entire grid.**

Keywords—cyber-physical security, anomaly detection, machine learning, smart grids, distributed energy resources, local outlier factor, distribution networks

I. Introduction

THE current electric grid has been designed for unidirectional power and data flow from large synchronous electric generators to consumers [1]. With the increased implementation of distributed energy sources (DERs), such as photovoltaic (PV) units, battery storage units, electric vehicles (EV), and EV chargers [2], the inevitable need for bidirectional power and data flow has risen. As a result, the current electric grid has been transitioning to Smart Grid (SG) to accommodate the continuously increasing penetration of DERs, which include renewable energy generation in the distribution network of the electric grid [1].

As with every new rising technology, benefits are accompanied by challenges that need to be identified, detected, and mitigated to ensure a trustworthy final product. With the increased need for communication flow and data exchange due to the implementation of DERs, new cybersecurity vulnerabilities arise. For example, false data attacks are a significant threat that cause severe consequences [3]. One type of false data attack could be false data injection, where the attacker would hijack (take control of) communication channels and lead to false state estimation and non-technical losses [3], such as energy theft, unmetered supplies errors, and conveyance losses [4]. One thing of utmost importance needs to be highlighted, which is the fact that cyberattacks can hinder grid operations that are governed by the laws of physics relating to the grid power flow, and not just information exchanges [6]. For instance, a data attack on the DERs, generators, or loads can interrupt power flow on the grid and result in severe physical damage [5]. As a result, a cyber-resilient power system that can detect and mitigate cyber-physical attacks is needed.

Previous literature has focused on developing algorithms to monitor grid behavior and detect discrepancies in grid performance due to cyberattacks. To detect non-technical losses, many papers have utilized supervised machine learning algorithms such as support vector machine (SVM), k-nearest neighbor (k-NN), decision tree (DT), regression-based, and artificial neural networks (ANN) [3]. Some papers used unsupervised learning algorithms, such as the gradient boosting classifier (GBC) and clustering-based algorithms [3]. Additionally, some papers have used deep learning techniques such as convolutional neural network (CNN) and a novel detection model called MFEFD [3]. Other authors have taken advantage of phasor measurement units (PMUs), which are measurement devices that can provide real-time data about the electric grid status [6]. Chamie et. al [6], who has used Micro-PMU measurements, focused on creating an anomaly detection algorithm that can detect data attacks in grid-edge devices. A pseudo-supervised learning (PSL) algorithm is used, where isolation forests is used as an unsupervised learner first, and then, nonlinear regressors is used as a supervised learner. Pandey et. al [7] created a detection algorithm that detects and classifies anomalies into three events: active power, reactive power, and fault events. Pandey et. al [7] has used statistics, clustering, maximum likelihood criterion (MLE), and density-based spatial clustering (DBSCAN) for event detection, and physics-based rule and decision tree for event classification. Deng et. al [8] concluded that one way to detect false data injection attacks in Smart Grids is to protect meter measurements from being manipulated by using PMUs.

As seen from the literature survey, much of the prior work

L. Al Homoud, R. Reghunath, S. Bayes, and R. S. Balog are with Texas A&M University at Qatar, Doha, Qatar (leen.alhomoud@ieee.org, rinith.reghunath@ieee.org, safin.bayes@ieee.org, robert.balog@ieee.org).

A. A. Peerzada and K. R. Davis are with Texas A&M University, College Station, Texas (aaqib.peerzada@ieee.org, katedavis@tamu.edu).

2021 North American Power Symposium (NAPS) | 978-1-6654-2081-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/NAPS52732.2021.9654478

has focused on detecting data attacks and mitigating consequences instead of detecting calculated, intelligent, and sophisticated cyber-attacks aimed at harming and disrupting the grid's cyber-physical security. Little work has been done to detect cyber-attacks while considering the grid's physical nature [3], [6], [8]. As a result, it is imperative to address the cybersecurity concerns holistically [6]. That can only be achieved if work is specifically done and focused on the cyber-physical consequences of attacks.

Therefore, in this paper, we propose an algorithm that enhances the cyber-physical resiliency of the grid. Our project aims to detect cyber-attacks on a smart distribution network that consists of various controllable assets, such as, capacitor banks, DERs, bus voltage regulators etc. We are focusing on sustained, intelligent, and coordinated attacks which can cause accelerated harm to the network's assets. Our detection scheme is based on a Machine Learning (ML) algorithm which will detect anomalies in the commands issued to the assets' controller.

II. Methodology

The methodology is divided in three phases: modeling, simulation, and machine learning. First, we modeled the IEEE 34 System including a Photovoltaic (PV) System and a storage element to simulate the effects of DERs. The complete network is shown in Figure 1. Then, we ran an annual simulation of the model to obtain the base case data, followed by running 100 Monte Carlo simulations, each time modifying the load profiles and the PV irradiance profiles. These formed the normal operating data. To obtain the malicious data, we issued various commands to the model and labeled those as malicious that caused harms to the system. The Local Outlier Factor (LOF) algorithm was used for anomaly detection. The algorithm was trained completely on the normal operating data and tested on both the normal and malicious data.

A. Modeling

1) Photovoltaic System

The PV system modeled in OpenDSS comprises of the PV array, an inverter, and a Norton Equivalent circuit. When a PV array collects solar energy, it transforms that solar energy into an electrical current that enters the inverter, entering the Norton Equivalent circuit to convert the current source into the required voltage source. The PV system was modeled with a maximum power point of 200 kW, making the PV penetration around 14%. The PV system was connected to bus 836 with a rated voltage of 24.9 kV, a rated apparent power of 500 kVA, a unity power factor, and a base irradiance and temperature of 1 kW/m^2 and 25°C, respectively. To model the inverter, we applied an inverter efficiency curve following data collected from the California Energy Commission [9]. These inverter efficiency values were for the ABB PVI-3.0-OUTD-S-US-A inverter, a 3 kW, 208 Vac grid support utility-interactive inverter with an arc detector. The temperature and irradiance profiles were obtained from the Qatar Environment and Energy Research Institute (QEERI). Both are yearly profiles based on data collected from 2017.

2) Storage Element

We also included a storage element as one of the Distributed Energy Resources (DERs) in our distribution network model. The storage element is a battery element that performs energy storage in the power grid. When the energy level drops in the grid, the storage element is activated and provides the needed power. Depending on the storage state, the storage element is connected to the grid differently. During the charging state, the storage element acts as a constant power load to the grid. During the discharging state, it acts as a generator that provides power to the grid. During the idle state, the storage element is modeled as being disconnected from the grid. We modeled the storage element in our distribution network to be connected to the network at bus 836, with a rated power of 1000 kW, a charge and discharge rate of 100%, and a rated storage capacity of 50,000 kWh.

3) Voltage Regulators

A voltage regulator is an electro-mechanical transformer that can vary its tap-ratio without interrupting the load and hence raise or lower the secondary-side voltage. There are two voltage regulators on the IEEE 34 System at buses 814 and 852. The rated voltages are 122 and 124 V, respectively

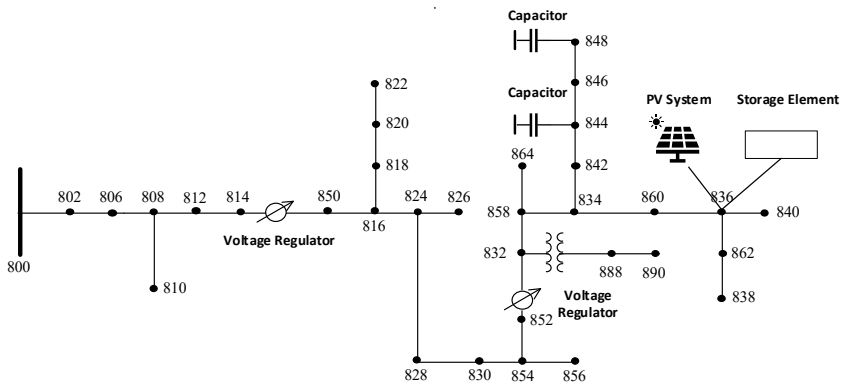

Fig. 1. Electrical distribution network one-line diagram which was based on the IEEE 34 bus test system with PV and energy storage added.

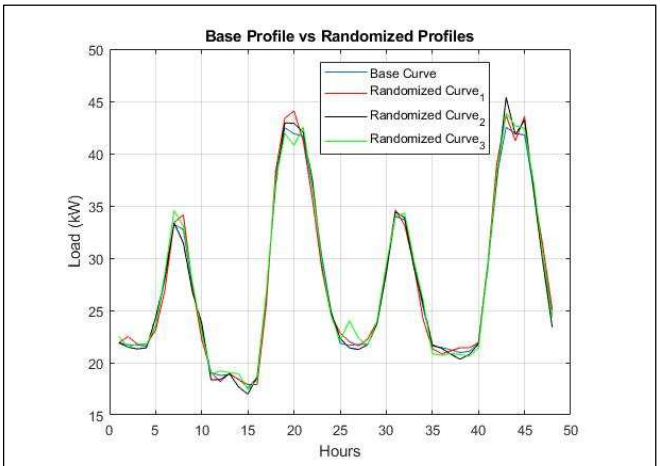

Fig. 2. Base and Randomized Load Profiles

for voltage regulators 1 and 2. Each voltage regulator consists of 32 tap positions, labeled from -16 to +16. By changing their tap positions in this range, they maintain a voltage within the specified bandwidth at the bus they are located in.

4) Capacitors

The two capacitors in the IEEE 34 System are situated at buses 846 and 848, respectively. The rated apparent power ratings of the capacitors are 300 and 450 kVARs, respectively. The capacitors have two thresholds, namely, "ONsetting" and "OFFsetting", where "Onsetting" is the bigger value. If the bus voltage is higher than the "ONSetting" the capacitor turns on. If it is below the "OFFSetting" the capacitor turns off. If the bus voltage is between the two thresholds the capacitor maintains its existing state.

5) Load

The loads are modeled based on the details provided by the IEEE 34 System datasheet [10]. However, there are no load profiles associated with the loads in the datasheet. To run a time-series simulation, OpenDSS requires a load profile for each load model to simulate the behavior of the load with respect to time. We used a webtool, called REopt Lite, developed by the National Renewable Energy Laboratory (NREL) [11], to obtain the load profiles. The tool generates an annual load profile in intervals of one hour for a given climate zone and building type. There are 16 climate zones in the webtool, and we chose the one represented by Phoenix, Arizona. This is because among all the representative cities, the weather of Arizona is similar to that of Doha, Qatar. Among the 17 building types, we picked five: Midrise Apartment, Hospital, Office, and Restaurant. The other building types yielded load profiles which have the shape like at least one of the aforementioned building types.

The load profiles are randomized at each Monte Carlo simulation by multiplying the values in the base profile with a scaling factor, that was obtained from a Burr Distribution. The parameters of the distribution are set such that the probability to obtain a scaling factor close to 1. By multiplying with a scaling factor close to 1, the points in the base profile are underestimated or overestimated, but not to an extent that it changes the entire shape of the profile. Figure 2 shows a graph of the base load curve plotted against the randomized load curves.

B. Simulation

1) Base Case

The base case represents the model's behavior during normal operating conditions. We require a base case data to serve as the reference dataset for the rest of the research. we run 100 Monte Carlo iterations of annual simulations where the time step in each annual simulation is one minute, corresponding to the time step of the irradiance and temperature profile of the PV system, which has the smallest time step resolution in the model. Thus, we have 525,600 time instants in one annual simulation. At each Monte Carlo iteration, the load profiles are randomized to account for the varying load demands. The irradiance profile is also randomized to account for the random occurrences of dust and cloud transients. The randomization is done in a similar manner to the load randomization.

The algorithm has two loops - an outer loop and an inner loop. The inner loop solves the OpenDSS model and obtains the data at each n^{th} time instant for the annual simulation. The inner loop stops execution when n = N = 525,600. The outer loop randomizes the load profiles and the irradiance profiles for each Monte Carlo iteration, "m". The outer loop stops execution when m = M = 100.

2) Running the Base Case

Load and irradiance randomization was performed to account for the varying behavior of the load and irradiance in each year. We connected our OpenDSS simulation with MATLAB to issue malicious control signals to the various assets of the distribution network and to save our circuit data more efficiently.

Depending on our usage of the network considering our project, we decided on collecting the following data from our network: bus voltages, capacitor states and reactive power, irradiance and real power of the PV system, tap positions of the voltage regulators for all the phases, storage capacity and states, power flowing into and out of the storage element, and sub-bus power.

3) Test Dataset

To create the test datasets, we would need to first issue malicious commands. The purpose of issuing malicious commands to the simulation is to observe how the system reacts to the malicious set points. This will help us identify the features that could be used in the machine learning algorithm and obtain the malicious data set. The malicious set is obtained in two steps. First, we issue arbitrary malicious commands to the assets and then identify the system characteristics that deviate from the base case behavior. Then, we issue several hundred random commands which are passed through a test based on our observations that label a command as anomaly or normal. We performed various test cases by issuing commands at arbitrary time instants and

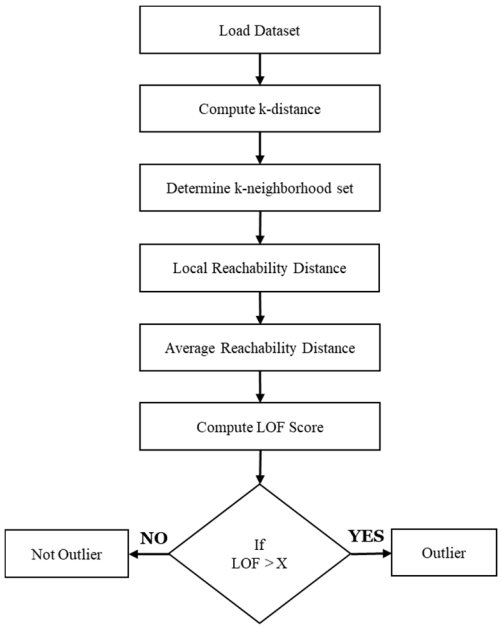

Fig. 3. LOF Algorithm Flowchart

observed changes on the other assets.

For the voltage regulators, we identified that the regulator tap positions and ANSI limit violations are the strongest indicators of a malicious command. In the next step, we issued random commands to the regulator and passed them through a test to label them as anomaly or normal. The command that was launched to the regulators was changing the tap position between -16 and 16.

To create the malicious dataset for the PV system, a similar approach to issuing the commands for the regulators was followed. The command that was launched to the PV system involved changing the percentage of PV production, which is also known as curtailment. To perform curtailment, different commands with different percentages of PV production were issued. we have concluded that both regulators are the two assets affected by issuing malicious commands to the PV system. As such, we started creating the malicious dataset for developing the anomaly detector for this asset. The steps followed are also quite similar to generating the malicious dataset for the regulators, where a random instant is first selected. Afterward, a command with a randomly selected percentage of PV production is launched at that random instant. A test is then run to determine if the command that was issued is malicious, which involves comparing the number of tap position switching in both regulators before and after launching the command. If the number of tap position switching is higher after launching the command, then the command is determined to be malicious and is used to create the malicious dataset that we will then use for the cross-validation and testing phases.

C. Machine Learning

1) Algorithm

For our anomaly detection algorithm, we are using an unsupervised learning method called Local Outlier Factor (LOF) algorithm. The LOF algorithm computes the local density deviation of a given point with respect to its neighbors. Each point in the dataset is given an LOF score and if the LOF score is greater than a certain threshold, the point is an anomaly. The LOF score is based on density calculations and therefore the samples that have a much lower density than their neighboring points are classified as outliers. To summarize the steps behind the LOF algorithm, at first, the algorithm calculates the reachability distance from all the neighbors (in the k-distance neighborhood of the given point) to the given point. Then, we proceed to calculate the local reachability distance of the point. The LOF score of the point is calculated using the local reachability of its neighbors and the point itself. If their LOF score is greater than the threshold value set, the point is considered an anomaly. These steps are then repeated for every point in the dataset. Our anomaly detector model assumes a developed anomaly detector for each of the three assets, where the assets communicate with each other to share their operating status. Figure 3 describes how our algorithm runs.

When running the tests to develop the malicious datasets for the regulators and the PV system, we were able to better understand our system, which helped us determine the features for each anomaly detector. For Regulator 1 (Bus 814) and Regulator 1 (Bus 852), the features are the time instant, the command issued, and the regulator tap positions in the previous time instant. For the PV system, the features are the time instant, command issued, PV production in the previous time instant, and regulator tap positions in the previous time instant.

Another thing that needed to be determined before training and testing the algorithms were the sizes of the training, cross-validation, and testing datasets. The training dataset contains the normal operating conditions of our model, which is also referred to as the base case data. The cross-validation and testing datasets of normal operating points and the malicious points that were obtained from the tests discussed earlier. The sizes of the training, cross-validation, and test datasets are around 3.6 million, 1 million, and 1 million, respectively.

2) Evaluation Metrics

After developing the anomaly detection algorithm, we needed to evaluate how well the algorithm does on detecting the malicious commands. For evaluating our anomaly detection algorithm, we utilize the method of F1-Score. We chose this evaluation metric over other common methods such as accuracy due to its performance in real-life scenarios.

The F1-Score is computed using precision and recall. Precision is the ratio of correctly classified anomalous samples to the total number of samples classified as anomalies. The value of precision is calculated as:

$$Precision\ (P) = (TP)/(TP+FP)$$

Recall is the ratio of correctly classified anomalous samples to the total number of anomalous samples in the dataset and is calculated as:

$$Recall\ (R) = (TP)/(TP+FN)$$

where,

- True Positive (*TP*) is the number of outcomes where the algorithm correctly classifies a point as an anomaly.
- False Positive (*FP*) is the number of outcomes where the operating point is not an anomaly but the algorithm incorrectly classifies the point as an anomaly.
- False Negative (*FN*) is the number of outcomes where the point is an anomaly but the algorithm incorrectly classifies the point as not an anomaly.

The F1-Score is computed using the following equation involving Precision (P) and Recall (R).

$$F1\ Score = (2*P*R)/(P+R)$$

The F1-Score gives us a value between zero and one. The algorithm is considered perfect if the F1-Score is 1, and it is considered a failure when the F1-Score is 0.

III. Results

A. Regulators 1 and 2

For the first regulator (situated at Bus 814), the size of the training dataset was 7 Monte Carlo deviations, where each deviation consists of 525,600 points (corresponding to one-year's set of data). Therefore, the training dataset size was around 3.6 million points. The malicious dataset comprised of 376 malicious points, which meant the number of anomalies or actual positives are 376. The cross-validation dataset comprised of these anomalous points, in addition to 1 Monte Carlo deviation (525,600 points making up the actual negatives). As such, the cross validation dataset size was 525,976 points. Similarly, the testing dataset comprised of 525,600 normal points (actual negatives) and 400 malicious points (actual positives). The cross-validation analysis was done through altering the threshold value for each run.

We utilized the optimum k-value of 10 along with our threshold value of 1.8, to begin the testing phase for our Regulator 1 anomaly detector. The results of the test are displayed in Table 1. Using this threshold, we were able to obtain only 22 false negatives and 147 false positives. The value of precision was 0.72, recall was 0.945, and the F1 score was 0.8173. These are acceptable values given the trade-off between false positives and false negatives discussed earlier. Therefore, we finalized these values as the parameters for our Regulator 1 anomaly detector.

For the second regulator (situated at Bus 852), we followed a similar approach as to the one taken for Regulator 1 to develop the anomaly detector. The size of the training dataset was 7 Monte Carlo deviations. Therefore, the training dataset size was around 3.6 million points. The malicious dataset comprised of 360 malicious points, that meant the number of anomalies or actual positives are 360. The cross-validation dataset comprised of these anomalous points, in addition to 1 Monte Carlo deviation (525,600 points making up the actual negatives). As such, the cross-validation dataset size was 525,960 points. Similarly, the testing dataset comprised of 525,600 normal points (actual negatives) and 359 malicious points (actual positives). The cross-validation analysis was done through altering the threshold value for each run.

TABLE I: Results

Asset	k-Value	Threshold	Precision	Recall	F1 Score
Reg. 1	10	1.8	0.72	0.945	0.8173
Reg. 2	10	1.8	0.8452	0.9582	0.8982
PV	15	2.0	0.3744	0.82	0.5141

Analysis of the cross-validation results for Regulator 2 reveals a similar conclusion as with Regulator 1, which is that there was a trade-off between false positives and false negatives. The best F1 score is obtained for a threshold of 1.9. However, due to the high cost of false negatives, the optimal threshold was selected to be 1.8, as it only had 15 false negatives. This meant that out of the total 360 malicious commands, it only missed 15 malicious commands. Moreover, it only had 63 false positives, meaning that out of the 525,600 normal points, it only identified 63 normal points as anomalies. Once again, the machine learning code concluded the optimal k-value to be 10, which was therefore used. We utilized the optimum k-value of 10 along with our threshold value of 1.8, to begin the testing phase for our Regulator 2 anomaly detector. The results of the test are displayed in Table 1. Using this threshold, we were able to obtain only 15 false negatives and 63 false positives. The precision was 0.8452, recall was 0.9582, and the F1 score was 0.8982. These are acceptable values given the trade-off discussed, and therefore, we finalized these values as the parameters for our Regulator 2 anomaly detector.

B. Photovoltaic (PV) System

For the photovoltaic system, we followed a similar approach as to the ones done for both the regulators to develop the anomaly detector. The size of the training dataset was 7 Monte Carlo deviations. Therefore, the training dataset size was around 3.6 million points. The malicious dataset comprised of 100 malicious points, which meant the number of anomalies or actual positives are 100. The cross-validation dataset comprised of these anomalous points, in addition to 1 Monte Carlo deviation. As such, the cross validation dataset size was 525,700 points. Similarly, the testing dataset comprised of 525,600 normal points (actual negatives) and 100 malicious points (actual positives). The cross-validation analysis was done through altering the threshold value for each run.

Upon analysis of the cross-validation results, we reached a similar conclusion as to the previous regulator where there was a trade-off between false positives and false negatives. We determined the optimal threshold to be 2.0, as it only had 16 false negatives. This meant that out of the total 100 malicious commands, it only missed 16 malicious commands. Moreover, it only had 137 false positives, meaning that out of the 525,600 normal points, it only identified 137 normal points as anomalies. Using this threshold value, the precision was 0.3801, recall was 0.84, and the F1 score was 0.5234. Our

machine learning code concluded the optimal k-value to be 15, which was therefore used. We utilized the optimum k-value of 15 along with our threshold value of 2.0, to begin the testing phase for our PV system. The results of the test are displayed in Table 1. Using this threshold, we were able to obtain only 18 false negatives and 137 false positives. The precision was 0.3744, recall was 0.82, and the F1 score was 0.5141. These are acceptable values given the trade-off discussed, and therefore, we finalized these values as the parameters for our photovoltaic system anomaly detector.

IV. Conclusion

Considering the increased implementation of distributed energy resources, there exists a high need for a change in the existing grid paradigm. Smart grids offer an intelligent, reliable, sustainable, economic and secure electric supply for the electric network. However, the grid might be subjected to various coordinated attacks, at the cyber or the physical level. Implementing a robust security system to defend against malicious activities is integral to the success of smart grids. Current anomaly detection techniques are focused on attacks that result in real-time consequences to the network. However, an intelligent, well-coordinated attack that doesn't cause immediate changes in network performance can bypass these detection techniques.

Our project offers a defensive framework against this by detecting the malicious commands issued to assets in a smart distribution network. We have developed a machine learning based anomaly detector capable of identifying a malicious command issued to the asset's controller. Our approach of detecting anomalies in the commands rather than anomalies in the monitored data enhances the system's cyber-physical resilience by detecting any command that does not make sense based on the historical patterns of the received commands and/or the status of the network. This enhances the integrity and improves the acceptance of the paradigm at the market, community, and socio-political level.

Acknowledgment

The authors would like to thank Engr. Sawsan Shukri for her helpful consultation to this research. The authors acknowledge and thank the Qatar Environment and Energy Research Institute (QEERI) for providing the irradiance and temperature datasets, from their outdoor test facility, that is used in this study.

References

[1] R. S. de Carvalho and D. Saleem, "Recommended functionalities for improving cybersecurity of Distributed Energy Resources," *2019 Resilience Week (RWS),* 2019.

[2] K. Shallenberger, "DER aggregation: Sector experts identify emerging trends in a nascent market," *Utility Dive*, 24 June 2017. [Online]. Available: https://www.utilitydive.com/news/der-aggregation-sector-experts-identify-emerging-trends-in-a-nascent-marke/447670/. [Accessed 15 August 2020].

[3] L. Cui, Y. Qu, L. Gao, G. Xie and S. Yu, "Detecting false data attacks using machine learning techniques in smart grid: A survey," *Journal of Network and Computer Applications*, vol. 170, 2020.

[4] SP Energy Networks, "Non Technical Loses," *SP Energy Networks*, [Online]. Available: https://www.spenergynetworks.co.uk/pages/non_technical_losses.aspx. [Accessed 10 August 2020].

[5] U.S Department of Energy, "Financial Opportunities: Funding Opportunity Exchange," 5 February 2020. [Online]. Available: https://eere-exchange.energy.gov/Default.aspx?Search=landscape&SearchType=. [Accessed 13 September 2020].

[6] M. El Chamie, K. G. Lore, D. M. Shila, and A. Surana, "Physics-Based Features for Anomaly Detection in Power Grids with Micro-PMUs," *2018 IEEE International Conference on Communications (ICC),* 2018.

[7] S. Pandey, A. K. Srivastava, and B. G. Amidan, "A real time event detection, classification and localization using Synchrophasor Data," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4421–4431, 2020.

[8] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, "False data injection on state estimation in power systems—attacks, impacts, and defense: A survey," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 411–423, 2017.

[9] "Grid Support Inverter List- Full Data", California Energy Commission, 2021. [Online]. Available: https://www.energy.ca.gov/media/2365. [Accessed: 25- Jan- 2021].

[10] IEEE Power Engineering Society, "Resources | PES Test Feeder," 1992. [Online]. Available: https://site.ieee.org/pes-testfeeders/resources/. [Accessed 13 September 2020].

[11] "REopt Lite | REopt Energy Integration & Optimization | NREL", Reopt.nrel.gov, 2021. [Online]. Available: https://reopt.nrel.gov/tool/. [Accessed: 25- Jan- 2021].