Attack-Resilient and Secure EMS: Design, Algorithms, Operational Protocols, and Evaluation

Final Project Report

S-72

Power Systems Engineering Research Center
Empowering Minds to Engineer
the Future Electric Energy System

Attack-Resilient and Secure EMS: Design, Algorithms, Operational Protocols, and Evaluation

Final Project Report

Project Team

Lalitha Sankar, Project Leader

Oliver Kosut

Arizona State University

Manimaran Govindarasu

Iowa State University

Graduate Students

Zhigang Chu

Roozbeh Khodadadeh

Andrea Pinceti

Jiazi Zhang

Arizona State University

Vivek Kumar Singh

Pengyuan Wang

Iowa State University

PSERC Publication 18-06

September 2018

For information about this project, contact:

Lalitha Sankar
Arizona State University
School of Electrical, Computer, and Energy Engineering
Engineering Research Center
551 E. Tyler mall, Room 585
Tempe, AZ 85281
Phone: 480-965-4953
Email: lalitha.sankar@asu.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
527 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-727-2052

Notice Concerning Copyright Material

Acknowledgements

We wish to thank Dr. Kory Hedman and his students from Arizona State University for their continuous support with the implementation and integration of software tools into the EMS platform.

We would also like to thank each and every one of our industry advisers. In particular we wish to thank the following people for their helpful suggestions and insightful comments: Evangelos Farantatos (EPRI), Benjamin Kroposki (NREL), Eugene Litvinov (ISONE), Harvey Scribner (SPP), and Mark Westendorf (MISO).

Executive Summary

During the past decade, the electric utility industry has increasingly used a combination of Energy Management System (EMS) and Supervisory Control and Data Acquisition (SCADA) systems to analyze, manage, and control the power system. The use of these monitoring and computing systems has helped power systems operators and utilities (stakeholders) run more efficiently, securely, and economically. Harnessing the processing power of modern computers and advances in telecommunication and information technology, these EMSs ensure that the systems operators have an accurate understanding of the network health and real-time status while ensuring wide-area control and situational awareness.

The increased reliance on information technology (IT) necessitates addressing the associated cybersecurity risks which are inherent to any cyber-physical system (CPS). The Department of Energy (DOE), Department of Homeland Security (DHS), and the North American Electric Reliability Corporation (NERC) have identified concerns that the grid is vulnerable to sophisticated coordinated cyber-attacks. With this project we addressed the following fundamental question: can reasonably realistic (i.e., attackers with limited capabilities) cyber-attacks be modeled and tested on electric power system (EPS) simulation platforms to evaluate: (a) attack severity and consequences, and (b) resiliency of energy management systems (EMSs) to such attacks?

In Part 1 of the project, the ASU team developed a software EMS platform on which sophisticated attacks and countermeasures were tested, validated, and demoed. The work of the ISU team in building a hardware testbed for cybersecurity studies is described in Part 2.

Part 1: Development of commercial grade energy management system and testing of cyber-attacks and countermeasures

The goal of this project is to identify realistic cyber-threats and vulnerabilities of modern EMSs as well as to propose countermeasures that system operators can implement in their systems. To obtain the most representative and accurate assessments possible, we developed a software platform which very realistically mimics the functions and operation of real world energy management systems. This java-based EMS platform, developed in collaboration with IncSys, includes a state estimator (SE) with bad data detector, real-time contingency analysis (RTCA), and security constrained economic dispatch (SCED). This complex tool allowed us to both verify how stealthy cyber-attacks are not detected by the EMS as well as observing the resulting physical consequences in a precise way. The platform has been used to observe the resiliency of systems of different sizes, varying from 200 to over 2000 buses, against a diverse number of attacks ranging from line overloads, to hiding post-contingency violations and stability limit violations.

Intelligently designed false data injection (FDI) attacks have been shown to be able to by-pass the state estimation bad data detector present in an energy management system (EMS) and to introduce arbitrary errors in the states of the system under attack. These false measurements can be computed so that they will create unobservable physical consequences, such as line overflows. As part of the analysis of the vulnerability of real power systems to these types of attacks, in this project we have

formulated four computationally efficient problems that can be used to design attacks on realistic, large scale systems.

The first two methods are based on row and column generation techniques which are useful to solve large linear programs which would otherwise be intractable. Row generation, which reduces the number of constraints, is used on line flow limits: we exclude all the constraints corresponding to lines which have a small power flow, as those are unlikely to be active in the optimal solution. If any of the constraints which have been omitted are violated in the final solution, they are added back to the problem and a new solution is calculated. The second algorithm uses both row generation and column generation. In this case, generators which are unlikely to modify their output as a consequence of the attack are removed from the problem, reducing the number of binary variables. The third method is based on the difference maximization algorithm, and in addition to being a linear program it also provides upper and lower bounds on the solution. The last algorithm uses Benders' decomposition, in which the problem is divided into two sub-problems which are solved iteratively.

These four algorithms have been tested on different test systems to compare their performance from a computational efficiency point of view as well as efficacy of the computed attacks. Moreover, the results have been used to assess the vulnerability of the systems. In general, every time a line is congested it is possible to create an attack which will lead to an overflow on such line. Lastly, the results are not only dependent on the level of congestion of the target line but also on the overall congestion of the system.

One of the assumptions of the attacks described is that the attacker has knowledge of system-wide information such as topology, generation data, etc. As we are looking at large scale test cases, it is reasonable to assume that it would be hard for an attacker to be able to gather full system data. For this reason, in this project we look at the possible physical consequences resulting from attacks designed with limited information. Specifically, we assume that the attacker has knowledge of only a small subgraph of the network and absolutely no knowledge of the outside system. This requires the attacker to be able to model the system response without any knowledge on its topology, generation, and loads; to this end, multiple linear regression can be used to predict the power injection at the boundary buses of the attack subgraph. This modified attack model is tested on the IEEE 24 and IEEE 118 bus systems, showing that successful attacks can be computed, even though the magnitude of the overflows is lower than the case in which the attacker has full knowledge of the system.

Network topology is important system data used in various data processing modules in the EMS. Changes in topology can result from either system incidents or malicious physical attacks; but, in general, such topology alterations can be detected in the cyber layer. However, a sophisticated attacker can launch cyber-attacks that alter the topology information in an unobservable manner; furthermore, they can also mask a physical attack via a cyber-attack to create a more coordinated attack. This type of unobservable cyber-attacks on topology can be of two types: line-maintaining and line-removing. For a line-maintaining attack, the attacker changes measurements and line status information to make it appear that line that is not in the system is now shown as active at the control center via SCADA data; the opposite is achieved by a line-removing attack. Using similar techniques as the attacks described above, we have shown that this type of coordinated

attacks is feasible and cannot be detected by current EMS technology. As per our results on the IEEE 24 bus system, large unobservable overloads of transmission lines can be achieved.

After having identified vulnerabilities of large scale systems that can be exploited by gaining access to the EMS, we focus our attention on possible countermeasures to protect the system from cyber-attacks. The FDI attacks described in the previous paragraphs create physical consequences by injecting false measurements from which the system operator gathers a wrong representation of the system loads. It is the mismatch between the real loads and the generation redispatch resulting from the fake loads which lead to the overloading of transmission lines. Based on this observation we designed three detectors, each relying on a different machine learning algorithm, to analyze and validate the observed loads in real time. The detectors are designed and trained to learn the patterns which exist between real system loads; when a new set of measurements is processed by the state estimator, the resulting loads are checked for the previously learnt patterns to understand if they represent normal or anomalous data. The three machine learning techniques used are: nearest neighbor, support vector machine, and replicator neural network. From the tests performed on the IEEE 30 bus system and on the synthetic Texas system, we have verified the high detection capabilities of the three detectors. Besides the very satisfactory performance of the algorithms designed, the importance of this approach lies in the fact that this countermeasure relies on data which is already being collected at the control center. Implementing these detectors in a real EMS would be fairly easy and accessible to any system operator.

Part 2: Anomaly detection for wide-area protection and control in smart grid

This research has focused on the design, development, and evaluation of robust algorithms for anomaly detection and mitigation for a couple of key applications in wide-area protection and control. The central theme of this research is to leverage Cyber-Physical System (CPS) properties of the grid and machine-learning algorithms to design robust Anomaly Detection System (ADS) that detects anomalies beyond the traditional information technology-based intrusion detection solutions. The following are the two key contributions. (1) The development of anomaly detection algorithm for Remedial Action Scheme (RAS), a wide-area protection scheme using a multi-agent based framework with machine-learning algorithms to distinguish attack behaviors from normal and fault behaviors, which in turn helps to achieve optimal protection decisions, such as load shedding. (2) The development of two anomaly detection algorithms for Automatic Generation Control (AGC): (a) model-based ADS and machine learning-based AGC. The former leverages the historical data as the redundant measurement to create different types of bounds/rules to detect attacks. The machine learning-based ADS deploys semi-supervised clustering algorithm for detecting cyber-attacks. The performance of the proposed anomaly detection and mitigation algorithms were evaluated through a combination of simulation and testbed-based experimental studies with realistic system models, datasets, and use-case scenarios. The results and testbed-based evaluations show that these algorithms achieve detection accuracies to the extent that they are ready for pilot deployment and evaluations. The research has also advanced the state-of-the-art research in CPS security for the power grid and the innovative use of machine-learning algorithms. This project also contributed to the workforce development in training several graduate students in research and also offering a couple of industry short courses for the utility industry.

Project Publications:

[1] J. Zhang, Z. Chu, L. Sankar and O. Kosut, "Can Attackers with Limited Information Exploit Historical Data to Mount Successful False Data Injection Attacks on Power Systems?," in *IEEE Transactions on Power Systems*.

[2] A. Pinceti, L. Sankar, and O. Kosut, "Load Redistribution Attack Detection using Machine Learning: A Data-Driven Approach", 2018 *IEEE Power & Energy Society General Meeting*, August 5-9, 2018, Portland, Oregon, USA, 2018

[3] Z. Chu, J. Zhang, O. Kosut and L. Sankar, "Evaluating power system vulnerability to false data injection attacks via scalable optimization," *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Sydney, NSW, 2016, pp. 260-265.

[4] Z. Chu, J. Zhang, O. Kosut and L. Sankar, "Vulnerability Assessment of Large-scale Power Systems to False Data Injection Attacks," [Online]. Available: https://arxiv.org/abs/1705.04218

[5] V. Singh, A. Ozen, M. Govindarasu, "A Hierarchical Multi-Agent Based Anomaly Detection for Wide-Area Protection in Smart Grid," Resilience Week 2018, August 20-23, Denver, Colorado, USA, 2018, (accepted for publication).

[6] K. Sarangan, V.K. Singh, M. Govindarasu, "Cyber Attack-Defense Analysis for Automatic Generation Control with Renewable Energy Sources," *North American Power Symposium (NAPS) 2018*, September 9-11, Fargo, North Dakota, USA, 2018, (accepted for publication).

[7] V.K. Singh, H. Ebrahem, M. Govindarasu, "Security Evaluation for Two Intrusion Detection Systems in Smart Grid SCADA Environment," *North American Power Symposium (NAPS) 2018*, September 9-11, Fargo, North Dakota, USA, 2018, (accepted for publication).

[8] P. Wang. M. Govindarasu, "Anomaly Detection for Power System Generation Control based on Hierarchical DBSCAN," *North American Power Symposium (NAPS) 2018*, September 9-11, Fargo, North Dakota, USA, 2018, (accepted for publication).

[9] V. Singh, M. Govindarasu, "Decision Tree Based Anomaly Detection for Remedial Action Scheme in Smart Grid using PMU Data," *IEEE Power and Energy Society 2018 General Meeting*, August 5-9, 2018, Portland, Oregon, USA, 2018.

Student Theses:

[1] Jiazi Zhang. *Vulnerability Analysis of False Data Injection Attacks on Supervisory Control and Data Acquisition and Phasor Measurement Units*. PhD Dissertation, Arizona State University, Tempe AZ, 2017

Part I

Development of commercial grade energy management system and testing of cyber-attacks and countermeasures

Lalitha Sankar
Oliver Kosut

Graduate Students
Zhigang Chu
Roozbeh Khodadadeh
Andrea Pinceti
Jiazi Zhang

Arizona State University

For information about this project, contact

Lalitha Sankar
Arizona State University
School of Electrical, Computer, and Energy Engineering
Engineering Research Center
551 E. Tyler mall, Room 585
Tempe, AZ 85281
Phone: 480-965-4953
Email: lalitha.sankar@asu.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
527 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-727-2052

Notice Concerning Copyright Material

Table of Contents

1. Introduction 1
   1.1 Background 1
   1.2 Overview of the problem 1
      1.2.1 EMS software platform 2
      1.2.2 Vulnerability of large scale systems 2
      1.2.3 Attack detection using machine learning 3
   1.3 Report organization 3
2. Simulation and demo platform 4
   2.1 Purpose and design 4
   2.2 Simulator structure 4
   2.3 Simulator software components 5
      2.3.1 Power analysis and network processing 5
      2.3.2 Security Constrained Economic Dispatch (SCED) 7
   2.3.3 Browser-based demo and simulation interface 7
   2.4 Demo of attacks and countermeasures 12
3. Vulnerability of large scale power systems to FDI attacks 14
   3.1 Motivation 14
   3.2 System and attack model 15
   3.3 Computational efficiency algorithms to solve attack optimization problems 16
      3.3.1 Row generation for line limit constraints 16
      3.3.2 Row and column generation for line and generator limit constraints 17
      3.3.3 Cyber-physical-difference maximization 17
      3.3.4 Modified Benders' decomposition for attacker-defender bi-level linear programs 17
   3.4 Simulation results 18
      3.4.1 Simulation methodology 18
      3.4.2 Computational efficiency 18
      3.4.3 Results on maximum physical power flows 19
      3.4.4 Results on attack resources 20
      3.4.5 Line vulnerability 21
      3.4.6 Impact of overall congestion 22

4. Limited information attacks 23
4.1 Motivation 23
4.2 Assumptions on attacker's knowledge and capability 23
4.3 Worst-case line overflow attacks with localized information 24
4.3.1 System power flow with localized information 24
4.3.2 Multiple linear regression to predict pseudo-boundary injections 24
4.3.3 Attack optimization problem under localized information 25
4.4 Simulation results 26
4.4.1 Simulation methodology 26
4.4.2 Results for IEEE 24-Bus RTS system 26
4.4.3 Results for IEEE 118-Bus system 28
4.4.4 Attack sensitivity to topology change 29
4.4.5 Verification of AC power flow model 29
5. Cyber-physical attacks 31
5.1 Motivation 31
5.1 System and attack model 31
5.1.1 System model 31
5.1.2 Attack model 32
5.2 Attack strategy 34
5.3 Numerical results 36
5.3.1 Solution for the attack designed with the attack strategy 36
5.3.2 Consequences of the Attack in the Nonlinear Model 37
6. Countermeasures 39
6.1 System model 39
6.1.1 Load data: model and design 39
6.1.2 Attack model and design 41
6.2 Detection algorithms 41
6.2.1 Nearest neighbor algorithm 41
6.2.2 Support vector machine 42
6.2.3 Replicator neural network 43
6.3 Experiments 44
6.3.1 Experimental methodology 44
6.3.2 Experimental results 45

7. Concluding remarks 48
References 49

List of Figures

Figure 1. Structure of the EMS with its main functional components 4
Figure 2. Detail of the state estimation block 6
Figure 3. Detail of the real-time contingency analysis block 6
Figure 4. Detail of the security constrained economic dispatch block 7
Figure 5. Demo simulator representation of the Cascadia network 8
Figure 6. Polish System as shown in the demo platform 9
Figure 7. Texas Synthetic Model 9
Figure 8. Injecting manually crafted bad data to test BDD functionality 10
Figure 9. Detected bad data in demo platform 10
Figure 10. Injecting attack data through state estimator configuration dialog box 11
Figure 11. Applying SCED set-points from cyber system to the physical system 11
Figure 12. Comparison between cyber RTCA and physical RTCA. The contingencies marked with *Current* are physical contingencies while *Saved* denotes a cyber contingency 12
Figure 13. Bi-level optimization problem 14
Figure 14. The maximal power flow vs. the $l1$-norm constraint (N1) with target (a) line 104, and (b) line 141 of IEEE 118-bus system. LS=10%. 19
Figure 15. The maximal power flow vs. the $l1$-norm constraint (N1) with target (a) line 24, (b) line 292, and (c) line 1816 of the Polish system. LS=10%. 20
Figure 16. (a) The maximal power flow and (b) $l0$-norm of the attack vector vs. the $l1$-norm constraint (N1) for target line 292 of the Polish system with different load shift. 21
Figure 17. The maximal power flow vs. the $l1$-norm constraint (N1) for target line 2110 of the Polish system. LS=10%. 22
Figure 18. The maximal power flow vs. the $l1$-norm constraint (N1) for target line 292 of the Polish system under different congestion levels 22
Figure 19. The space of cyber-attacks 23
Figure 20. Bilevel optimization problem under limited information 25
Figure 21. IEEE 24-bus RTS system decomposed into attack sub-network and external network 27
Figure 22. The maximum power flow (PF) v.s. the $l1$-norm constraint (N1) when target line is 28 of IEEE 24-bus system for (a) Scenario 1, and (b) Scenario 2 historical data. 27
Figure 23. The pseudo-boundary power injection error v.s. the l1-norm constraint (N1) when target line is 28 of IEEE 24-bus system for (a) Scenario 1, and (b) Scenario 2 historical data. ... 27
Figure 24. The maximum power flow (PF) v.s. the $l1$-norm constraint (N1) when target line is 5 of IEEE 118-bus system for (a) Scenario 1, and (b) Scenario 2 historical data. 28

Figure 25. The pseudo-boundary power injection error v.s. the $l1$-norm constraint (N1) when target line is 5 of IEEE 118-bus system for (a) Scenario 1, and (b) Scenario 2 historical data.... 28
Figure 26. Comparison of the maximum power flow of DC and AC attacks. 30
Figure 27 Temporal Sequence of Data Processing Units in The Cyber Layer within Attack. 32
Figure 28 *Time sequence of attack and system events.* 34
Figure 29 *Summary of all 38 target lines under attack* 37
Figure 30 Power flow variation on line 12 during 20 system events..... 38
Figure 27. Graphical representation of the concept of nearest neighbor 42
Figure 32. Graphical representation of the concept of support vector machine 43
Figure 33. Model of the replicator neural network used..... 44
Figure 34. Distribution of nearest neighbor distance for normal and attacked cases 45
Figure 35. Distribution of SVM scores for normal and attacked cases. Note that for SVM a higher score (closer to 1) represents a belief that the data is normal, whereas a lower score (closer to -1) represents attacked data..... 45
Figure 36. Distribution of replication error from the replicator neural network detector for normal and attacked cases..... 46
Figure 37. Receiver operating characteristic of the three detectors with 10% load shift attacks . 46

List of Tables

Table 1. Comparison of four proposed algorithms 16
Table 2. Comparison of average number of binary variables 18
Table 3. Statistics of computation time with 10% load shift 18
Table 4. Summary of the attack sub-network in IEEE 118-bus system 28
Table 5. Summary of the sensitivity analysis results under different topologies 29
Table 6. Summary of the sensitivity analysis results under AC power flow historical datasets .. 30
Table 7. Physical and cyber data for attack and system events 34
Table 8. Relative size of PJM zones and 30-bus system loads 40
Table 9. Performance of the detectors with 15% LS attacks 47

1. Introduction

1.1 Background

Recent successful cyber-attacks on secure federal databases, secure financial systems, and even on credit card databases that were seemingly designed to be safe and off-limits from hackers and cyber-attacks suggest strongly that electric power systems (EPS) may not be immune to intelligent attackers either. While one cannot protect a system against all-knowing "omniscient" attackers, most real-world successful cyber-attacks are not due to attacker's knowledge of the entire system but due to their ability to exploit simple and often known security weaknesses. The first level of defense for any cyber-physical system (CPS) including the electric power CPS should be to protect the physical and information technology infrastructure via a variety of physical (e.g., access control, substation protection, etc.) and cyber (e.g., firewall policies, intrusion detection, anti-virus software, and database security, to name a few) security mechanisms as recommended by NERC CIP standards. However, the cautionary tale to take away from recent attacks is that one must assume all systems can be hacked into. Therefore, there is a need to focus on second and third lines of defense. In particular, for EPS, since hacking in itself is insufficient to affect system operations, one must focus on credible data integrity threats that can affect operations and lead to systematic failures. Identifying such data integrity threats is an extremely important second line of defense; however, equally important is the crucial third line of defense that is often the most challenging: enabling real-time detection and protection against attacks. This proposal addresses both the second and third lines of defense. We assume that EPS communications and computer networks can be hacked into, when and where accessible from outside, given that such is the case for almost all other secure cyber systems. Identifying credible threats, evaluating their consequences, and developing defense mechanisms requires a detailed, realistic, and tractable model of electric power CPS operations; such a model should be inclusive of the numerous built-in resiliency mechanisms and the interactions between various system modules and grid operators.

1.2 Overview of the problem

After a malicious actor breaks into an EMS, a trivial type of attack would be to simply disconnect circuit breakers or otherwise change the operating condition of network devices without proper safety checks which could lead to blackouts or equipment damage. This type of attack has been carried out multiple times in the past decade against multiple utilities with the attack on Ukraine power grid system being the most prominent one [1]. The attack on the Ukrainian power grid clearly demonstrates that a SCADA system can be compromised. Based on this observation, we focus on a class of sophisticated, undetectable attacks.

In our previous work we have shown that a class of false data injection (FDI) attacks against state estimation (SE) can be both unobservable to system operators and at the same time cause physical consequences on the grid [2]. In this project we focused on three main issues:

1. Developing a realistic EMS software platform;
2. Studying the vulnerabilities of large scale systems to FDI attacks;
3. Developing countermeasures against FDI attacks.

1.2.1 EMS software platform

While much work on the study of cyber-attacks against the electrical grid can be found in the literature, often very simplified models are used and the simulation of the attack consequences lacks realism. To validate our attacks and identify realistic vulnerabilities of state-of-the-art Energy Management Systems we have developed a Java-based software platform which mimics all the main functions of an EMS. This tool is based on open source software for power system studies called OpenPA. Leveraging the basic functions of OpenPA we built different blocks to perform all fundamental EMS operations: state estimation (SE), real-time contingency analysis (RTCA), and security constrained economic dispatch (SCED). This platform allowed us to precisely observe the efficacy and the consequences of the cyber-attacks we created as well as to improve the attack strategies. Moreover, it made it possible to identify new vulnerabilities of the EMS which can be exploited by attackers.

1.2.2 Vulnerability of large scale systems

As previously stated, the integration of the communication cyber layer to power systems makes them vulnerable to cyber-attacks which could result in serious physical consequences or even system failure. Therefore, it is crucial to develop techniques to detect and thwart potential attacks, which requires evaluating system vulnerability to credible attacks. Assessing and evaluating consequences of possible attacks is extremely instructive for system operators, and is important for the secure operation of the power systems.

Optimization problems have been proposed to design FDI attacks that aim to maximize line power flow [2], maximize operating cost [3], or change locational marginal prices [4]. However, the results have only been demonstrated for small systems. Similar to [2], we consider an optimization problem to determine the worst-case FDI attack that causes line overflow, but our goal is to design optimization algorithms that scale to significantly larger systems (i.e. thousands of buses).

In addition to developing efficient ways to study the vulnerability of large scale systems, we also investigate the possibility of launching FDI attacks when the attacker has limited knowledge and access to the system. We assume the attacker only has access to information inside an attack sub-network and absolutely no knowledge of the outside network. In order to overcome the limited information, we suppose that the attacker infiltrates the sub-network long before it executes its attack, so that it can observe the natural behavior of the system in order to predict the effect of an attack. In particular, we assume that the attacker has access to historical data inside the sub-network that includes loads, costs, capacities, status, and dispatches of generators, and locational marginal prices. We suppose that the attacker uses multiple linear regression method to learn the relationship between the external network and the attack sub-network from historical data. Furthermore, we predict the response of the control center under such attacks in a local sub-network via a bi-level optimization problem.

The last type of attacks studied in this report involve the hiding of physical attacks on transmission lines via FDI cyber-attacks. We show that a coordinated cyber-physical attack in which a line is physically taken down while a cyber-attacks spoofs its status as sees by the system operator can lead to even greater line overloads and system disruption.

1.2.3 Attack detection using machine learning

One of the main goals of the project is the design of new resiliency mechanism that can be implemented into EMSs to enhance their performance and reliability under cyber-attacks. To this end, we have developed advanced detection algorithms based on machine learning techniques which can validate the incoming data and identify measurements which could possibly be maliciously modified. As the underlying effect of the FDI attacks is to create a false representation of the system loads, the detectors are designed to learn patterns within the loads and in real-time validate the observed measurements by searching for their hidden patters. In addition to the high detection rate of the mechanisms we have designed, the power of this approach lies in the fact that it leverages data which is already available to every system operator. This makes our solution easy to be adopted by utilities to improve the current bad data detectors.

1.3 Report organization

The following report is organized in five main sections. Section 2 describes the development of the java-based EMS platform, illustrating its main components and how it can be used for validation and demonstration purposes. In Section 3 computationally efficient algorithms for the design of FDI attacks are described, with a focus on their application to the study of system vulnerabilities. The problem of limited information attacks is introduced in Section 4, while the description of cyber-physical attacks is in Section 5. The design of countermeasures is described in Section 6. Section 7 concludes the report by summarizing the main results achieved with this project and by presenting possible ways in which this work could be used by utilities and the industry at large.

2. Simulation and demo platform

2.1 Purpose and design

Cyber-physical attacks involve both IT infrastructure and the physical system. As a result, simulating the attack and being able to demonstrate its effect on the physical system is of great practical importance. To achieve this, we have designed and implemented a simulation and demonstration platform.

2.2 Simulator structure

The following figure shows the general structure of an energy management system under false data injection attack.

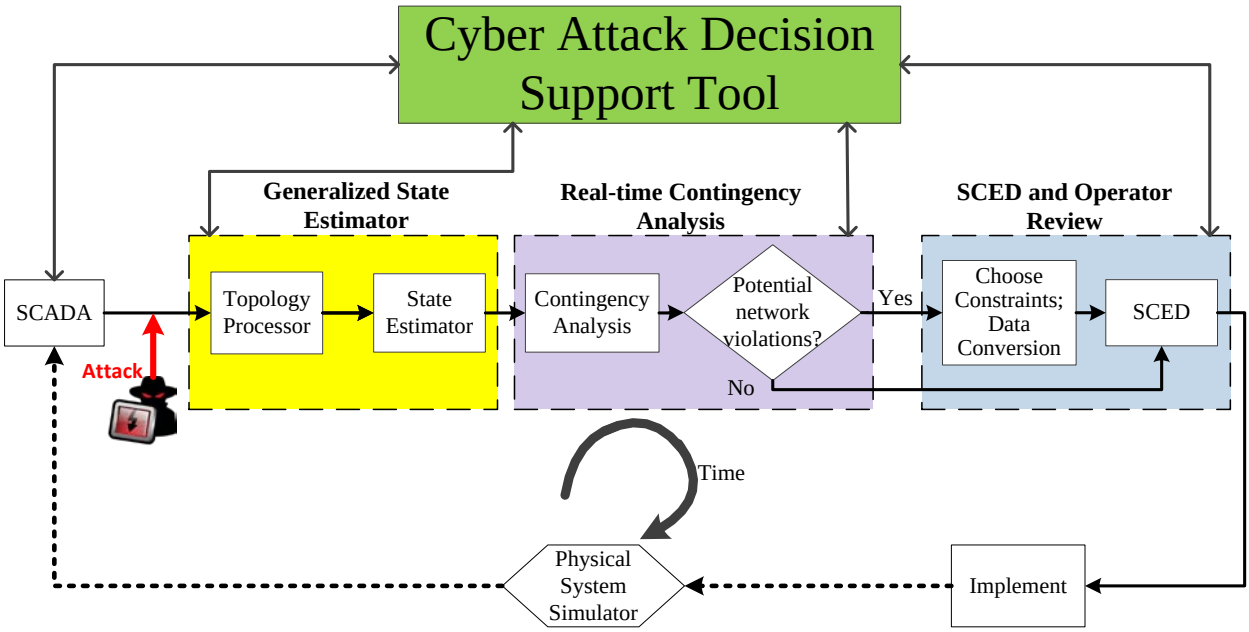


Figure 1. Structure of the EMS with its main functional components

A simulator needs to emulate all stages of the EMS as well as the attacker's actions. SCADA system can be simulated with an AC power flow. To simulate in a realistic manner the measurements collected by the SCADA system used as input for the state estimator, random noise is added to selected power flow results. Attacker action can also be simulated at this stage by replacing the true measurements with false measurements. Load estimator is simulated as part of the state estimator. The simulator then does a Contingency Analysis (CA) and a Security Constrained Economic Dispatch (SCED) to determine the dispatch for the next cycle of the EMS. In the next section, the software used to simulate each of these components is described.

2.3 Simulator software components

2.3.1 Power analysis and network processing

The demo platform uses OpenPA library as its backbone to run and calculate power grid state with a given input. OpenPA is an open-source library developed by IncSys in Java programming language. It provides the following computational packages.

2.3.1.1 OpenPA core

OpenPA Core provides model representation and conversion functions to well-known power system software such as PSS/e and PSLF. Model element data is stored using comma separated values (CSV) format which makes manipulation of the input more accessible and straightforward. OpenPA Core stores all the model data in memory in a lightweight object named PAModel. PAModel in turn is based on LINKNET structure for representing networks [5]. PAModel provides a reliable foundation to extract data required for network analysis and handle special cases that may arise during power grid operation (e.g. branch disconnections which lead to creation of new islands in the grid)

2.3.1.2 Power flow

Power flow module implements a fast-decoupled power flow (FDPF) algorithm. It utilizes OpenPA core (along with mathematical tools library) to calculate adjacency matrix, energized islands, B′, and B′′ matrices. The algorithm has sub-routines to distribute network loss among generators based on their relative capacity to absorb those losses. The module also gives user the ability to control the amount of slack distribution, iteration count of the algorithm, and flat-start option among others.

In general, power flow module provides a fast, reliable, and configurable way for the users to run power flow on an electric network and update the network elements with the data resulting from the power flow. This allows the use of power flow module both as a tool to simulate physical model and as a computational component inside the EMS.

2.3.1.3 State Estimation (SE)

State estimator determines the best estimate of the actual system state by solving an overdetermined power flow problem [6]. It uses a model of power system and whatever measurements are available [6]. The inputs to the state estimator are the network topology and network measurements. The output from state estimator is the estimated state (voltage magnitude and angle) and estimated measurements (branch flows and bus injections).

OpenPA provides a state estimation module that does state and load estimation as well as observability analysis and bad data detection (Figure 2). To solve the equations for large system,

OpenPA depends on Suitesparse library (SuiteSparse, n.d.). OpenPA Core also provides the support to correctly model which network devices are telemetered and how precise the obtained measurements are.

The bad data detector module uses an inverse Chi-Square test to determine if the state estimation solution contains bad data or not. In the case of bad data existence, the *bad* measurements are the ones with largest normalized residue error. OpenPA state estimator will identify observable and unobservable measurements, too.

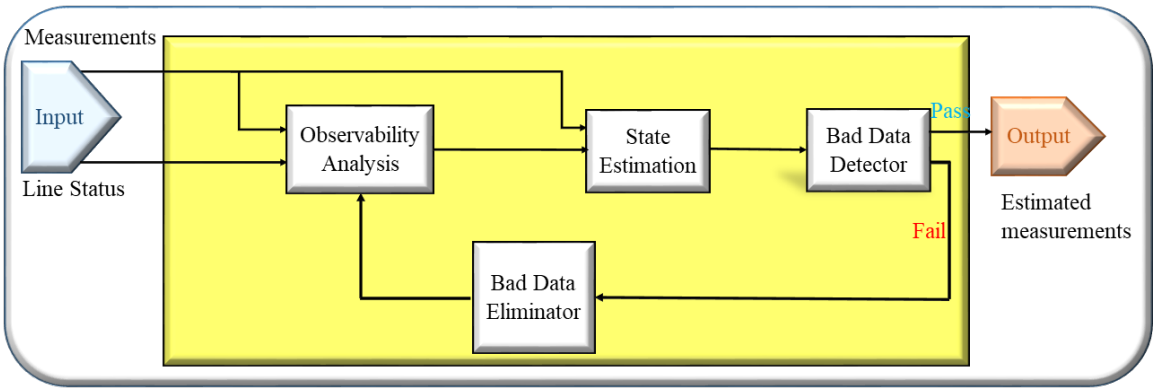


Figure 2. Detail of the state estimation block

2.3.1.4 Real-Time Contingency Analysis (RTCA)

Contingency analysis is used to compute system state after an outage. OpenPA provides a highly configurable and efficient RTCA module. Both the outages to be considered and the reported contingencies can be configured prior to running the RTCA program. For the purposes of the project, only non-radial branch outages are considered. OpenPA RTCA can also run in parallel mode. If a sufficient number of processing cores are available, it is capable of simulating thousands of contingencies in a matter of seconds. A schematic diagram of the RTCA is shown in Figure 3.

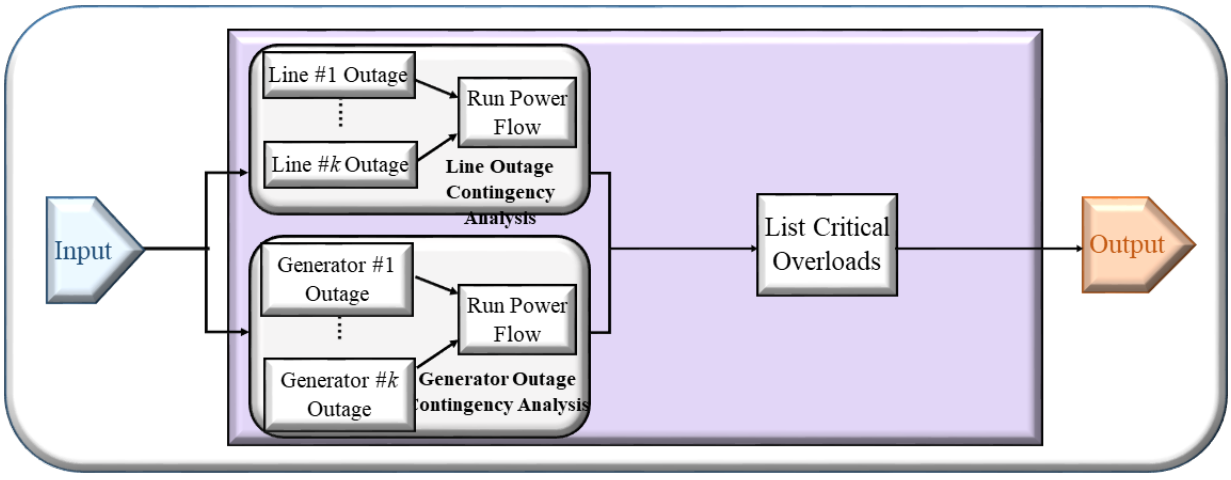


Figure 3. Detail of the real-time contingency analysis block

2.3.2 Security Constrained Economic Dispatch (SCED)

SCED package (Figure 4) has been developed in-house. It uses Mixed Integer Linear Programming (MILP) to find the optimal economic dispatch for the network. The package uses Gurobi™ as solver and it is written to be compatible with OpenPA. SCED builds an independent model of the system using data from OpenPA Core and RTCA outputs. Then it uses Gurobi™ to solve the model and acquire the optimal setpoints for all the generators in the system.
The SCED package has also been designed to be easily configurable. It has multiple options for network loss modeling and in case of model infeasibility, a number of slack variables can be activated to help analyze the roots of infeasibility.

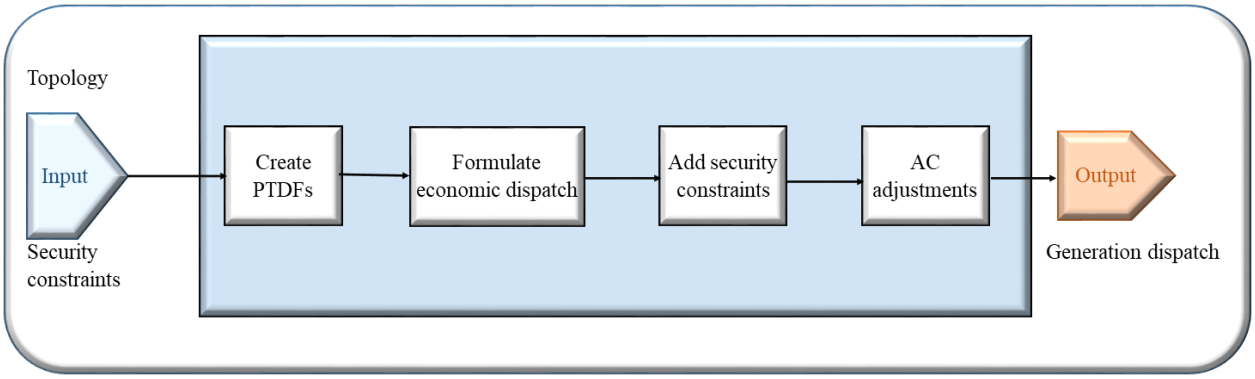


Figure 4. Detail of the security constrained economic dispatch block

2.3.3 Browser-based demo and simulation interface

The visual interface is a web application. Its backend has been developed using Play Framework [7] which is an open source web application framework. To visualize the network, originally D3js library [8] was used. Figure 5 shows how the demo simulator depicts a synthetic model (named Cascadia). This model is designed to be geographically compatible with Washington State. The attack is designed to hide contingencies on the four transmission lines connecting western and eastern parts of the state.
In the background, the framework utilizes OpenPA and SCED to run power flow, state estimation, RTCA, and SCED and updates the displayed graph based on the resulting network state.

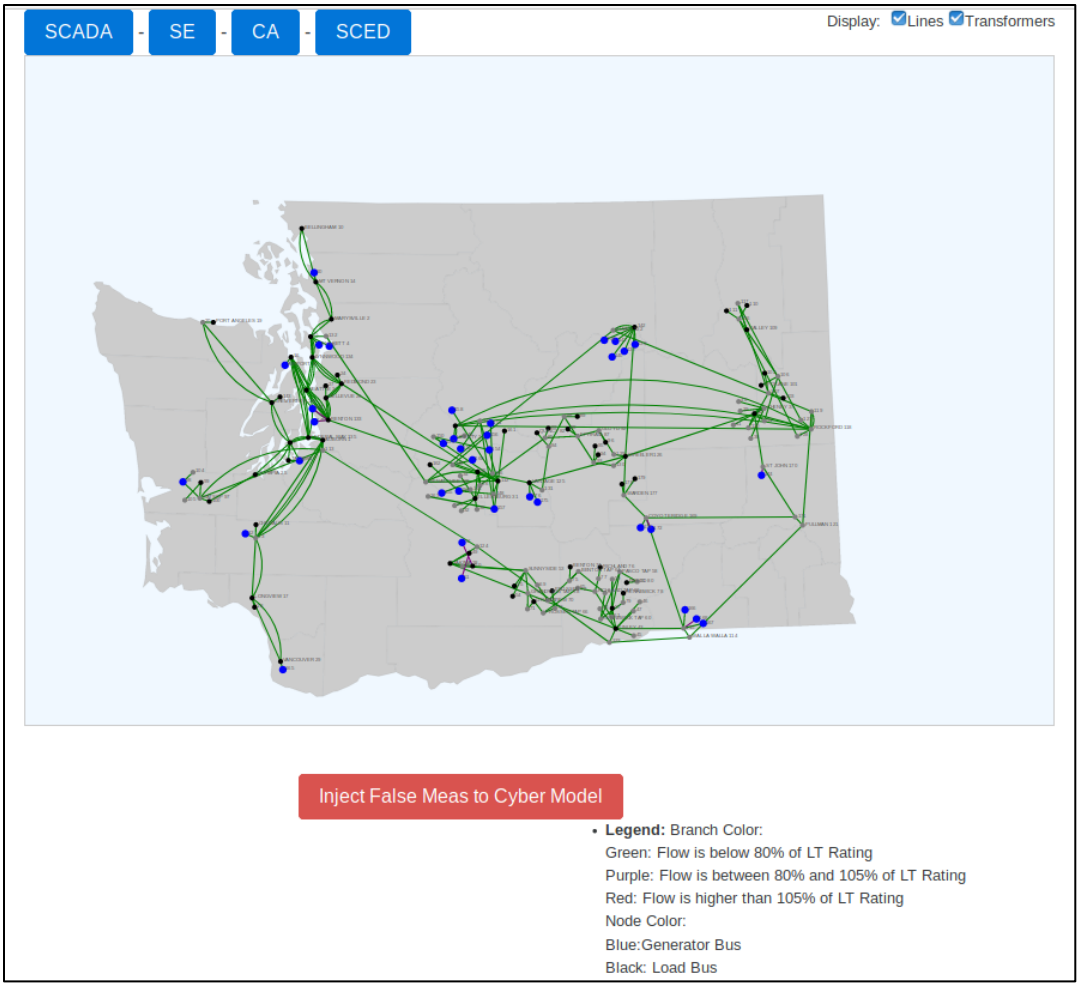


Figure 5. Demo simulator representation of the Cascadia network

Another attack was designed and carried out on Polish system. Because of the larger number of buses in that system and lack of geographical information, the visualization library was changed from D3js to Cytoscape.js [9]. Another contingency hiding demo visualization is shown in Figure 6 and Figure 7 using the updated visualization library.

The new platform has the capability of visualizing all available OpenPA case formats as well as providing configuration options graphically to the user. Figure 6 shows how the platform visualizes the Polish system. Figure 10 shows how the user can simulate attacker action by just clicking a checkbox. Selecting this option, the system automatically feeds attack measurements to the state estimator resulting in false output load estimates by the SE without any bad data flag.

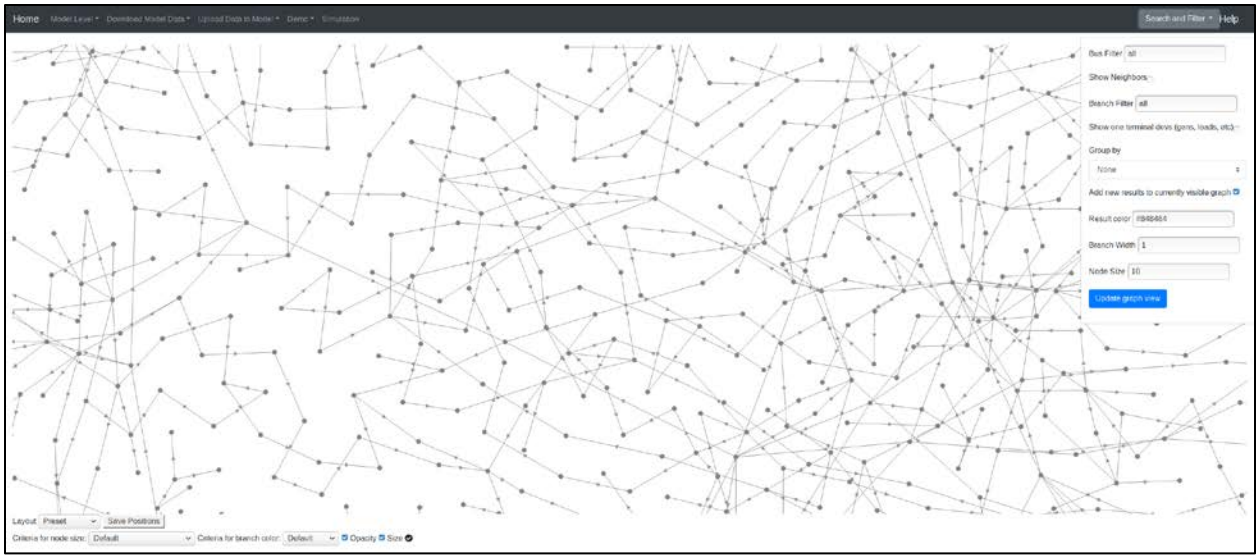


Figure 6. Polish System as shown in the demo platform

As another example, Figure 7 shows how the platform displays Texas Synthetic model.

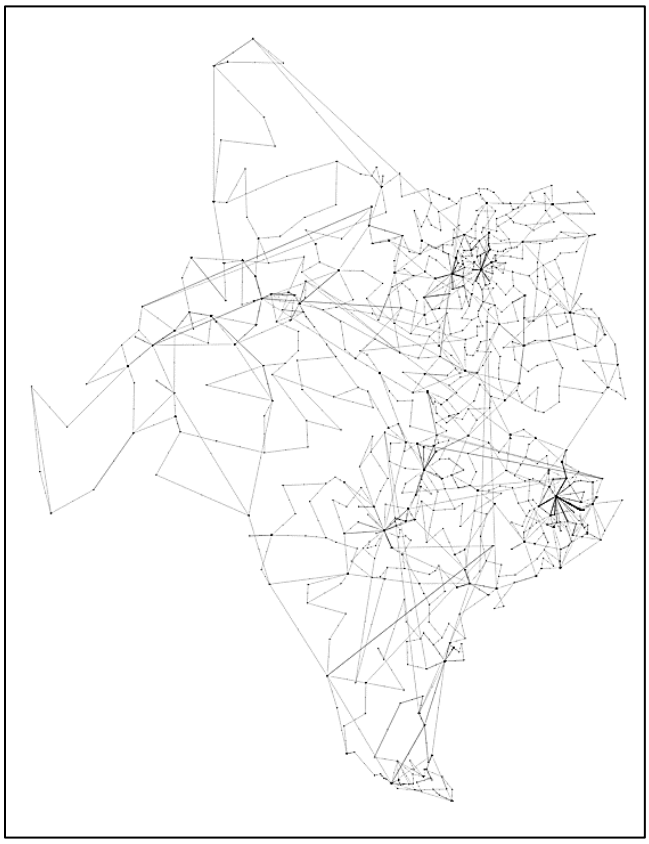


Figure 7. Texas Synthetic Model

To show that the bad data detector module does its job correctly, a few erratic measurements are programmed into the demo as Figure 8 shows.

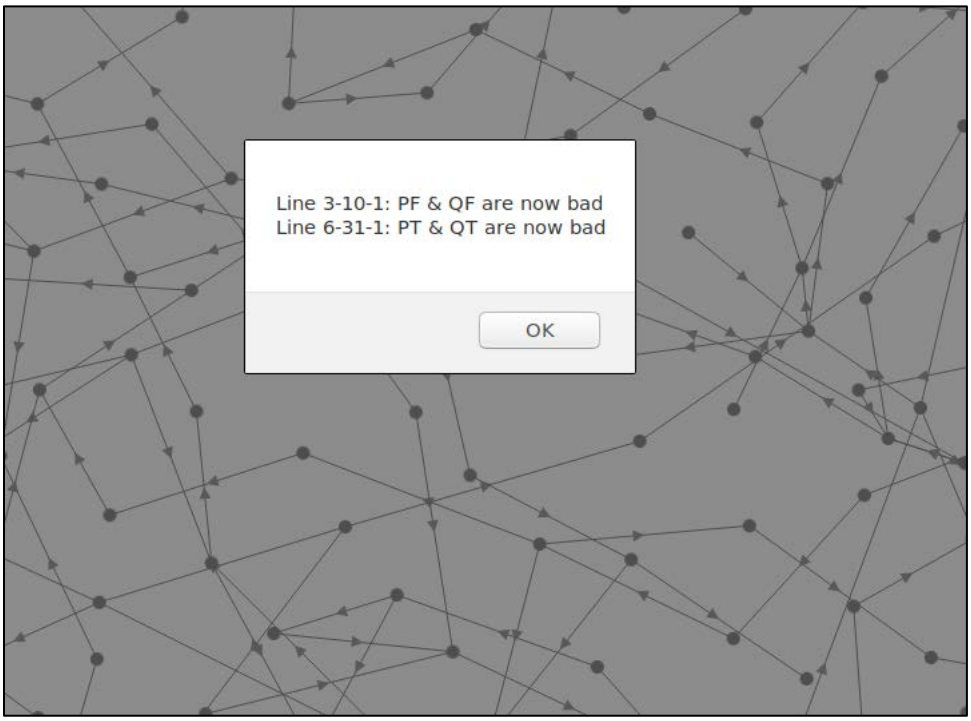


Figure 8. Injecting manually crafted bad data to test BDD functionality

After running state estimation, the following table is displayed (Figure 9):

ID	Type	Estimated Measurement	Telemetry measurement	Sigma
In-3-10-1	BranchFromQ	-10.8932	140	0.01
In-3-10-1	BranchFromP	-122.6769	-250	0.01
In-6-31-1	BranchToP	469.7069	100	0.01
In-6-31-1	BranchToQ	17.1143	-90	0.01

Figure 9. Detected bad data in demo platform

The platform strives to make all OpenPA functions available to the user through its easy-to-use interface. The user can craft their own bad data and inject it using the upload functionality. However, the user can use the ones already programmed into the demo to save time during a live presentation.

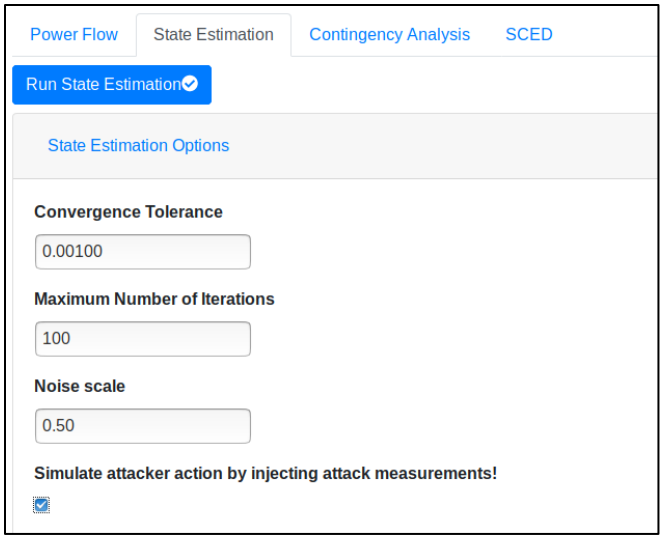


Figure 10. Injecting attack data through state estimator configuration dialog box

The user can simulate operator action during attack by selecting the proper tab from the EMS dialog box. The user can temporarily save RTCA results after RTCA execution for later comparisons. The program also automatically saves SCED dispatch results.

Assuming SCED dispatch suggestions are applied to the network by the operators, simulating what is happening in the physical system can be done by completely reloading the model and loading the dispatch from memory as the screenshot in Figure 11 shows.

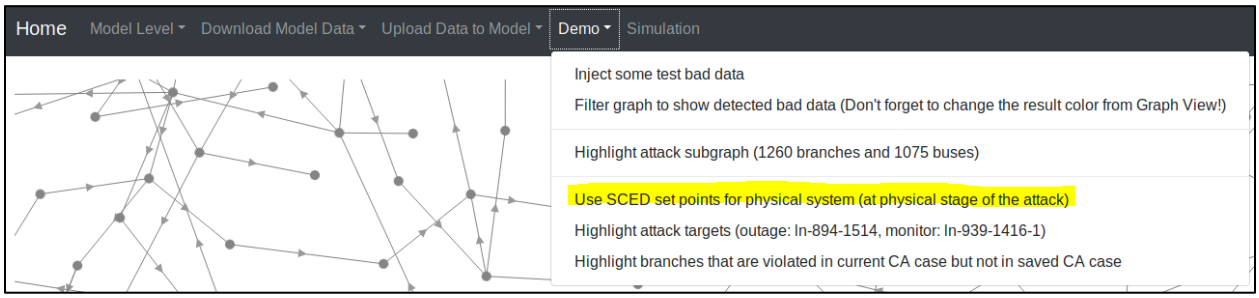


Figure 11. Applying SCED set-points from cyber system to the physical system

Running an RTCA after this action is equivalent to a contingency happening in physical system (albeit with a dispatch designed by the malicious actor). The platform provides the option to compare the saved RTCA (from cyber model) with the current RTCA (representing actual contingencies in the physical system). A screenshot of such comparison is shown in Figure 12.

Active	Status	Case	Contingency Branch	Monitored Branch	Pre Cont. Flow	Post Cont. Flow	Overflow Percent	Severity
☑	Current	Post Cont.	ln-1650-2077-1	ln-1930-2074-1	67.6	111.4	107.63	Violation
☑	Current	Post Cont.	ln-1781-2077-1	ln-1930-2074-1	67.6	111.33	107.57	Violation
☑	Saved	Post Cont.	tx2-2157-157-1	ln-2121-2380-1	14.72	84.98	105.56	Violation
☑	Current	Post Cont.	tx2-2157-157-1	ln-2121-2380-1	14.55	84.91	105.48	Violation
☑	Current	Post Cont.	ln-32-36-1	tx2-540-23-1	82.51	191.68	104.17	Violation
☑	Current	Post Cont.	ln-894-1514-1	ln-939-1416-1	115.09	166.94	103.69	Violation
☑	Current	Post Cont.	ln-1781-1920-1	ln-1930-2074-1	67.6	107.25	103.62	Violation
☑	Current	Post Cont.	ln-894-1514-1	ln-939-1513-1	113.27	164.3	102.05	Violation
☑	Saved	Post	ln-894-1514-1	ln-939-1416-1	113.29	162.76	101.09	Violation

Figure 12. Comparison between cyber RTCA and physical RTCA. The contingencies marked with *Current* are physical contingencies while *Saved* denotes a cyber contingency

It can clearly be seen that the contingencies in the physical world are different than the contingencies the cyber world knows about. This means the network is NOT operating in N-1 secure conditions but the operators will think so based on the false information they get from the compromised EMS.

2.4 Demo of attacks and countermeasures

The here described platform has been used to present and demonstrate our work on the design of attacks as well as countermeasures. In the previous section, the demo interface is described by showing a line overloading attack, which is one of the attacks we have tested on our platform. Depending on the specific system under study and its level of congestion, these attacks can be designed to create and hide either base case overloads or line violations resulting from contingencies. This class of attacks, which is the focus of this report, is an example of how false data injection attacks can have physical consequences that may harm the power systems. However, FDI attacks can be designed to target system states, system topology, generator dynamics, and energy markets. Optimization problems have been proposed to design FDI attacks that aim to maximize line power flow, change locational marginal prices, or maximize operating cost. Once the optimization problem is formulated, the algorithms introduced in the reports can be readily used to evaluate system vulnerability (i) on large-scale power systems; (ii) under limited information attacks.

The countermeasures described in section 5 should be implemented in addition to the normal EMS operations, and we are currently working on implementing it on the Java-based EMS platform here described.

3. Vulnerability of large scale power systems to FDI attacks

3.1 Motivation

In [2], an FDI attack against SE that leads to an overflow is introduced. The system operator re-dispatches the generation subsequent to the attack, resulting in an overflow on a target line. A bi-level optimization problem, shown in Figure 13, is formulated to model such attacks in which the first level models the attacker while the second level models the system response. The first level of this optimization problem is the attacker's problem, modeling its objective to maximize the power flow on a target line, subject to its limitations on: (i) resources to change states, characterized by the number of center buses in c; and (ii) detectability, characterized by the *load shift*, or the difference between the cyber (*i.e.* false) load and the original load, as a percentage of the original load. The second level is the system response to the attack, modeled by a Direct Current (DC) Optimal Power Flow (OPF).

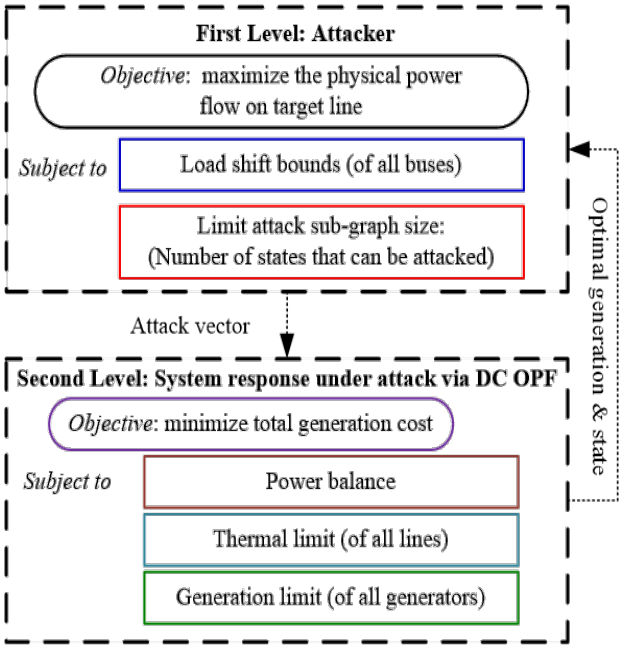


Figure 13. Bi-level optimization problem

Techniques to solve bi-level programs with applications to power systems have been studied in [3] [3],[10],[11], but the problems they study have the same objective for both levels, and hence their techniques cannot be applied to our problem. This bi-level optimization problem can be reformulated to a mathematical program with equilibrium constraints (MPEC) [12][13] by replacing the second level by its Karush-Kuhn-Tucker (KKT) conditions. However, MPECs are non-convex and hard to solve efficiently in general. Many heuristics have been applied to MPECs involving reformulations and relaxations [14]-[16], but they typically require non-linear programs and/or proprietary solvers. Moreover, they do not guarantee optimality, convergence, nor speed. We have attempted to apply existing techniques to this MPEC (*e.g.* the SNOPT solver with Matlab interface), but in our experiments they failed to produce good solutions even on small-scale

problems. In [2], this MPEC is further reformulated to a mixed-integer linear program (MILP) by rewriting the complementary slackness constraints as mixed-integer constraints. As the system size scales, the MILP formulated in [2] becomes harder to solve because the number of constraints and their associated binary variables increases with the size of the network.

To overcome this difficulty, we introduce four computationally efficient algorithms to evaluate the vulnerability of systems. In some cases, these algorithms yield the optimal attack. In cases in which it is intractable to find the optimal attack, these algorithms provide lower and/or upper bounds on the objective value. A lower bound highlights a specific overflow vulnerability for the system since it represents a feasible attack. An upper bound, on the other hand, constitutes a limit on the severity of this class of attacks.

The first algorithm yields the optimal attack by reducing the number of line limit constraints and their associated binary variables using row generation [17]. The second algorithm further reduces the number of binary variables by judiciously eliminating generation limit constraints using column generation [17]. However, optimality cannot be guaranteed by this approach, so the algorithm only provides a lower bound on the objective. The third algorithm provides both lower and upper bounds via a linear program (LP) that maximizes the difference between target line cyber and physical power flows. The fourth algorithm provides a lower bound on the objective by utilizing Benders' decomposition [18] to solve the original bi-level attack optimization problem instead of the re-formulated MILP. This algorithm not only applies to the original bi-level attack optimization problem, but also applies to any attacker-defender bi-level linear programs (ADBLPs).

3.2 System and attack model

The attacker's knowledge and capabilities are described as follows:

1. The attacker is able to perform system-wide DCOPF.
2. The attacker controls measurements in a subset S of the network.

An attack is defined to be unobservable if, in the absence of noise, there exists an $n_b \times 1$ attack vector $c \neq 0$ such that the measurement $\bar{z}$ modified by the attacker satisfies $\bar{z} = z + Hc$ [19]. An attacker with control of the measurements in S can launch this attack if Hc has non-zero entries only in S. Let $\hat{x}$ be the estimated states without attack. The residual $r = \bar{z} - H(\hat{x} + c) = z + Hc - H(\hat{x} + c) = z - H\hat{x}$ is the same as the residual without the attack. Therefore, this attack can bypass the DC bad data detector (BDD).

For tractability reasons, we use DC power flow model, but the attacks introduced in this paper can also be used to create false data that bypass AC BDD as in [2]. Given an attack vector c, unobservable false measurements can also be created even if AC measurement model and AC SE are used.

For an attack vector c, load buses (i.e., buses with load) corresponding to non-zero entries of c are denoted center buses. Given an attack vector c, a subgraph S bounded by load buses can be

constructed as in [20] that, if controlled by the attacker, can execute the unobservable attack associated with c. By modifying measurements only in *S*, the attacker can arbitrarily spoof the states of center buses without detection. The results of this unobservable attack will be seen by the system operator as load changes at load buses within *S*, while the total load of the system remains unchanged; it is for this reason that this class of attacks is also called *load redistribution attacks*. We model the power system with $n_b$buses, n_{br} branches, n_g generators, and n_m measurements.

3.3 Computational efficiency algorithms to solve attack optimization problems

To overcome the computational difficulties brought on by a large number of binary variables, four computationally efficient algorithms are introduced in this section. Table 1 lists the key features of all four algorithms, in addition to the original MILP.

Table 1. Comparison of four proposed algorithms

Algorithm	Program type	Outcome	Tractable test cases
Original MILP	MILP	Optimal solution	24-bus
Row generation for line limit constraints (RG)	MILP	Optimal solution	24-bus, 118-bus
Row and column generation for line and generator limit constraints (RCG)	MILP	Lower bound	24-bus, 118-bus, Polish (2383-bus)
Cyber-physical-difference maximization	LP	Lower & upper bounds	24-bus, 118-bus, Polish (2383-bus)
Modified Benders' decomposition for bi-level linear programs (MBD)	Iterative LP	Lower bound	24-bus, 118-bus, Polish (2383-bus)

3.3.1 Row generation for line limit constraints

Row and column generation techniques are useful in solving large-scale linear programs. For constraints of the form $Ax < b$, row generation retains only a subset of constraints (rows of A), and column generation retains only a subset of variables (columns of A). We iteratively add only those constraints and variables that are needed [21][22]. These techniques help reduce the size of matrix A, and hence accelerate the solving process. Similar techniques have been used by power system operators for large-scale optimization problems, including unit commitment and security constrained economic dispatch (SCED) [23]. In our problem, these techniques allow us to reduce the number of binary variables. If a line is operating with very low power flows compared to their ratings, its rating, the corresponding line limit constraint is unlikely to be active in the optimal solution of the original MILP, and therefore, can be removed. If the cyber power flow of a line is beyond its limit, we say this line has cyber overflow. If there are any post-attack cyber overflows, the line limit constraints for those lines are added back to the attack optimization problem (new

rows generated). If this algorithm terminates, the solution is guaranteed to be optimal because no constraints are violated.

3.3.2 Row and column generation for line and generator limit constraints

RCG modifies RG by further reducing the number of binary variables, now focusing on generator limit constraints. Since load changes are limited by the load shift factor, it is likely that in response to these load changes, only a small number of generators (denote as marginal generators) will re-dispatch. RCG reduces the number of binary variables associated with generator limit constraints by assuming the generation output of non-marginal generators remains unchanged after attack. Similar to RG, post-attack cyber overflow lines are added to the set of critical lines (new rows generated). In addition, generators whose post-attack generation are different from pre-attack values are added to the set of marginal generators (new columns generated). This ensures the system response to the attack predicted by the attacker is correct. RCG can be efficiently applied to the Polish system with 2383 buses since the number of binary variables is significantly reduced. Since some of the variables are held constant, RCG is not guaranteed to yield the optimal solution for the original MILP. However, it does provide a feasible solution (lower bound).

3.3.3 Cyber-physical-difference maximization

The DM algorithm maximizes the post-attack power flow difference between physical and cyber power flows. Both lower and upper bounds can be derived using DM. Moreover, this algorithm only involves a linear program, and hence, can be applied efficiently on large systems. The DM algorithm solves an optimization problem which maximizes the difference between cyber and physical power flows on target line, subject to the attacker's constraints. The resulting attack vector can be injected into a DCOPF to yield a lower bound on the physical power flow of the target line. The resulting maximum cyber-physical-difference can be used to calculate an upper bound on the physical power flow of the target line by adding it to the line limit.

3.3.4 Modified Benders' decomposition for attacker-defender bi-level linear programs

Benders' decomposition [24] can be utilized to solve a linear program with complicating variables in a distributed manner at the cost of iteration [25]. It is a popular technique to solve optimization problems of large size or with complicating variables. It is also effective in solving complex optimization problems such as stochastic programs and mixed-integer linear programs. In Benders' decomposition, an optimization problem is decomposed into two sub-problems, wherein variables of each sub-problem are treated as constant in the other. The two sub-problems are solved iteratively until the solution converges. The MBD algorithm is formed by modifying the classic Benders' decomposition algorithm to apply it on any ADBLP without converting it into an MILP. An attacker-defender bi-level linear program can be converted to a single-level problem by replacing the defender's problem with its primal-dual optimality conditions. The resulting single level problem is non-convex and hard to solve because of a bilinear term in the constraints. Benders' decomposition is utilized to decompose this problem into two problems, where each of them treat the variable of the other problem as constants. The slave problem uses the optimal solution of the master problem as inputs, and its dual variables at solution are used back in the master problem for optimality cuts.

3.4 Simulation results

3.4.1 Simulation methodology

In this section, we present numerical results using the algorithms described in Section 3.3. Two test systems are used, namely the IEEE 118-bus system and the Polish system. As stated in Section 3.3, we note that the convergence of RG is not computationally tractable for the Polish system in a reasonable length of time. Therefore, we only apply RCG, DM and MBD to the Polish system. Prior to the attack, the IEEE 118-bus system and the Polish system have 7 and 17 critical lines (lines with power flow over 90% of their limits) and 15 and 6 marginal generators, respectively. The vulnerability of these two systems are evaluated exhaustively by targeting all critical lines. The l_1-norm constraint N_1 is chosen in the range [0.1:1] for the 118-bus system and [0.1: 2] for the Polish system, both with increment 0.1. Throughout, Matlab, Matpower, and the Gurobi solver are used to perform the simulations. All tests are conducted using a 3.40 GHz PC with 32 GB RAM.

3.4.2 Computational efficiency

The decrease in the number of binary variables characterizes the computational efficiency improved by RG and RCG. Table 2 illustrates a comparison of the average number of binary variables when applying the original MILP, RG, and RCG on both test systems. For the Polish system, the number of binary variables of RG is an estimate, as we are unable to verify the convergence of RG in a reasonable length of time. This table demonstrates that both RG and RCG can greatly reduce the number of binary variables compared to the original MILP, and therefore significantly improve the computational efficiency. Table 3 illustrates the statistics of the computation time for several target lines using the proposed algorithms with 10% load shift. For each target line, each algorithm is tested for the full range of N1 values stated above. We note that RCG is more efficient than RG since it requires fewer binary variables. As one would expect, DM, an LP, is the most efficient among the four proposed algorithms. Note that the number of iterations for MBD varies for different parameter choices (target line, N1, and LS), resulting in a large variation in computation time.

Table 2. Comparison of average number of binary variables

Test system	Original MILP	RG	RCG
118-bus	480	122	45
Polish	6446	688	87

Table 3. Statistics of computation time with 10% load shift

Target line	Algorithm	Max (s)	Min (s)	Avg (s)	Med (s)
37 of 118-bus	RG	7.53	0.95	3.33	1.9
	RCG	1.25	0.34	0.76	0.69
	DM	0.5	0.43	0.47	0.45
	MBD	1.88	1.57	1.63	1.59
24 of Polish	RCG	46.36	3.40	20.39	13.67
	DM	15.75	1.91	8.09	8.58

	MBD	12.26	10.46	11.4	11.58
292 of Polish	RCG	76.34	27.47	39.29	33.69
	DM	16.77	1.91	7.02	6.10
	MBD	1846.2	9.86	358.73	10.31

3.4.3 Results on maximum physical power flows

Figure 14 illustrates the maximal power flow on target lines 104 and 141, respectively, when applying the four different algorithms described in Section 3.3 on the IEEE 118-bus system with 10% load shift. We use this system to compare the bounds found by RCG, DM and MBD to the exact solution provided by RG; note, however, that RG is intractable for the Polish system. RCG provides the optimal solution for all target lines we have considered in the 118-bus system. Figure 14. The maximal power flow vs. the l_1-norm constraint (N1) with target (a) line 104, and (b) line 141 of IEEE 118-bus system. LS=10%.

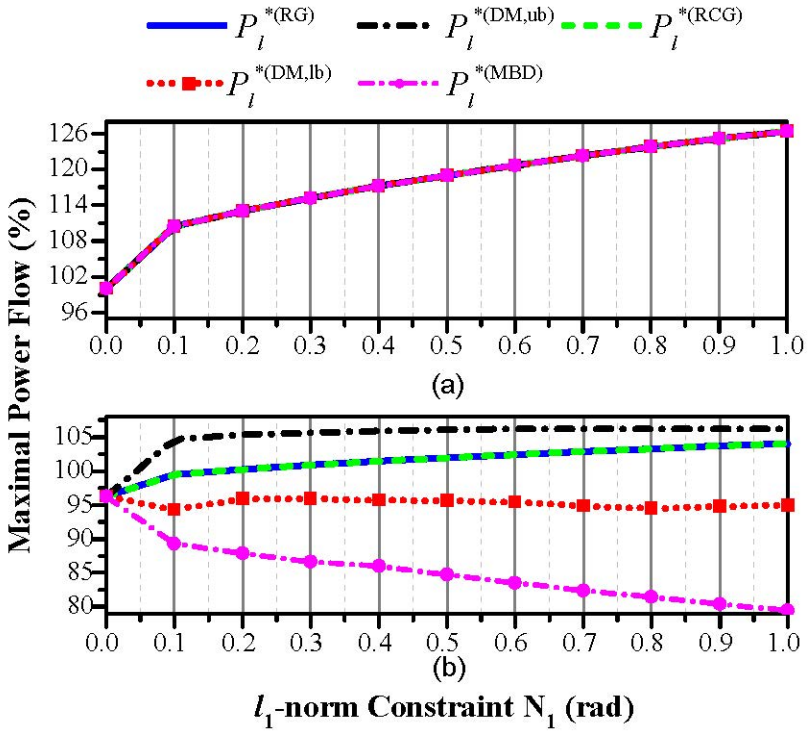


Figure 14. The maximal power flow vs. the l_1-norm constraint (N1) with target (a) line 104, and (b) line 141 of IEEE 118-bus system. LS=10%.

Results for target lines 292, 24, and 1816 of the Polish system are illustrated in Figure 15. The load shift constraint is 10%. For target line 292, all four algorithms yield the optimal solution in the range $N_1 \epsilon$ [0.1: 1.6]. For the remaining N1, our algorithms do not give the optimal solutions. We observe that the upper and lower bounds do not match for target line 24, but MBD yields the tightest lower bound. For target line 1816, DM provides the tightest lower bounds.

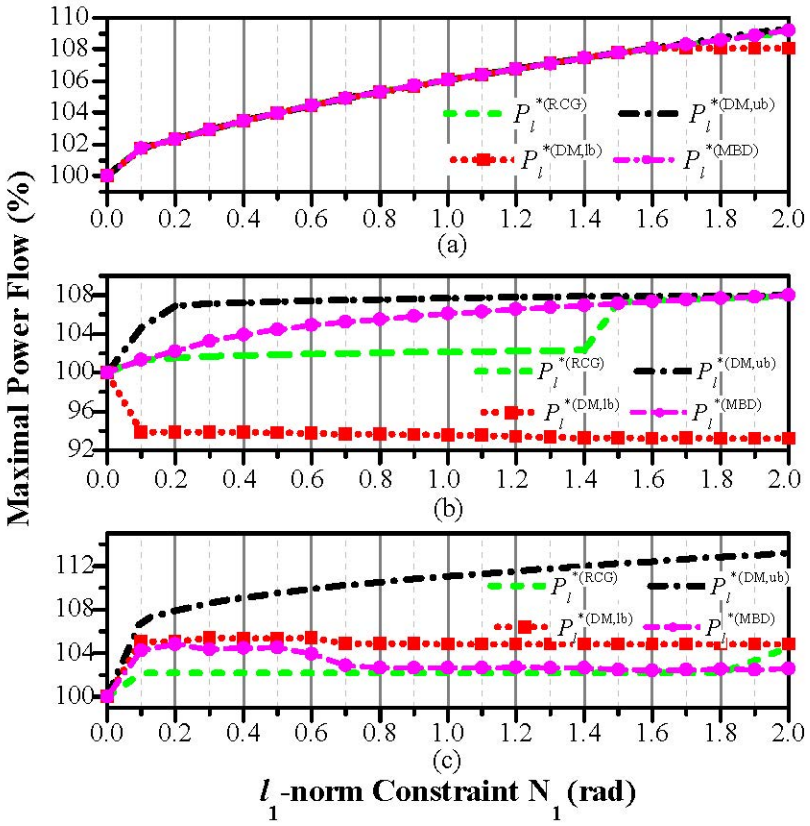


Figure 15. The maximal power flow vs. the l_1-norm constraint (N1) with target (a) line 24, (b) line 292, and (c) line 1816 of the Polish system. LS=10%.

3.4.4 Results on attack resources

Figure 16 illustrates the relationship between maximal power flow and l0-norm of the attack vector (i.e. the number of center buses in the attack) versus the l_1-norm constraint N1 for target line 292 of the Polish system, with different load shift constraints. As N1 increases, so does the l_0-norm of the attack, indicating that l_1-norm is a valid proxy for l_1-norm for our problem. If a larger load shift is allowed, the maximal power flow on target line increases, but the resulting l0-norm decreases. This indicates a trade-off between load shift and attacker's resources: as the attacker attempts to avoid detection by minimizing load changes, it will require control over a larger portion of the system to launch a comparable attack. Similar results are also obtained on the IEEE 118-bus system.

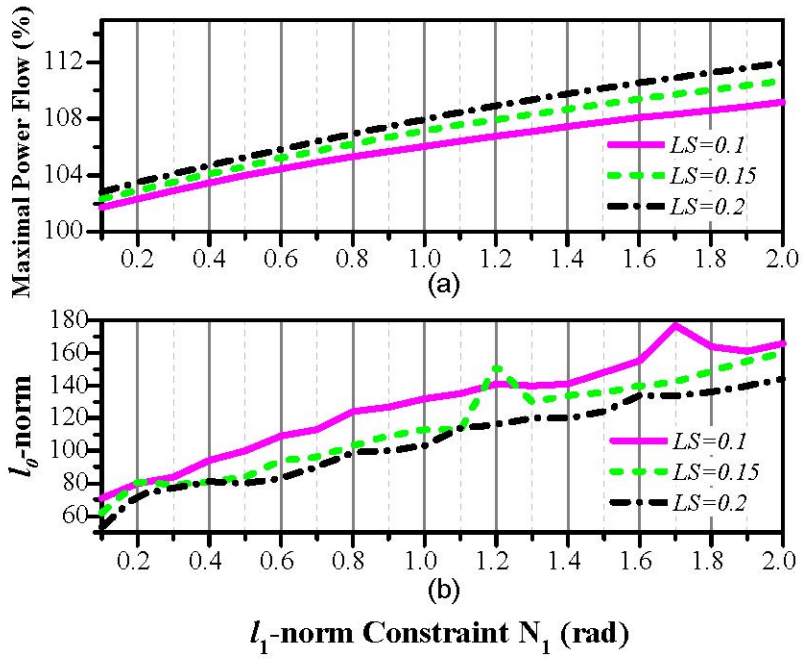


Figure 16. (a) The maximal power flow and (b) l_0-norm of the attack vector vs. the l_1-norm constraint (N1) for target line 292 of the Polish system with different load shift.

3.4.5 Line vulnerability

Since the objective of the attack is to maximize the physical power flow on a target line, it is intuitive that congested lines are more vulnerable to this attack. We have found experimentally that almost every congested line can be overloaded. One exception is line 176 in the IEEE 118-bus system. This is because line 176 is a radial line: it is the only line connected to a bus with a generator and no load. The line limit constraint in the OPF ensures that no possible dispatch could cause the line power flow to exceed the limit, even if based on counterfeit loads. In fact, any line with this radial configuration is immune to the proposed attack; moreover, these radial lines represent the only exceptions to our finding that congested lines can be overloaded. We have also found that lines that are not congested pre-attack may still be vulnerable to this attack, such as line 141 in the IEEE 118-bus system (Figure 14 (b)) and line 2110 in the Polish system (Figure 17).

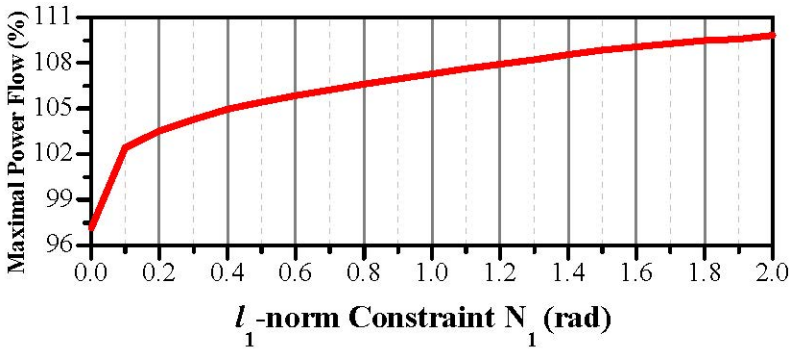


Figure 17. The maximal power flow vs. the l_1-norm constraint (N1) for target line 2110 of the Polish system. LS=10%.

3.4.6 Impact of overall congestion

In the above, we have shown that virtually all critical or congested lines are vulnerable to overload. However, the extent of the vulnerability depends on several factors, such as the overall congestion of the system. This phenomenon is illustrated in Figure 18, which shows the worst-case attack for line 292 of the Polish system under different overall congestion levels. This overall congestion is adjusted by uniformly changing the line ratings for all lines. Note that higher line ratings mean a less congested system. As shown in Figure 18, as the overall congestion level increases, the maximal power flow on the target line also increases, even though the line is equally congested before attack in each case.

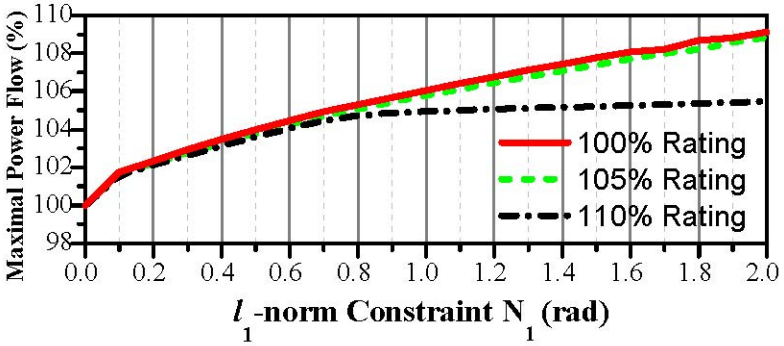


Figure 18. The maximal power flow vs. the l_1-norm constraint (N1) for target line 292 of the Polish system under different congestion levels

4. Limited information attacks

4.1 Motivation

Unobservable attacks leading to severe physical consequences can be designed via a bi-level optimization problem as shown in the previous section. This attack optimization problem requires the attacker to know system-wide information including topology, generation cost and capacity, and load data. In practice, obtaining all the required information can be difficult for the attacker. In order to ensure that the worst-case attacks are credible, we focus on understanding whether it is possible at all to design FDI attacks with only limited system information. Recently, [26]-[28] have demonstrated that it is possible to design FDI attacks against SE with inaccurate or limited topology information. However, physical consequences of the worst-case limited information FDI attacks have not been analyzed. The space of cyber-attacks is shown in Figure 19, and in this task we focus on the attacks categorized in the red shaded area.

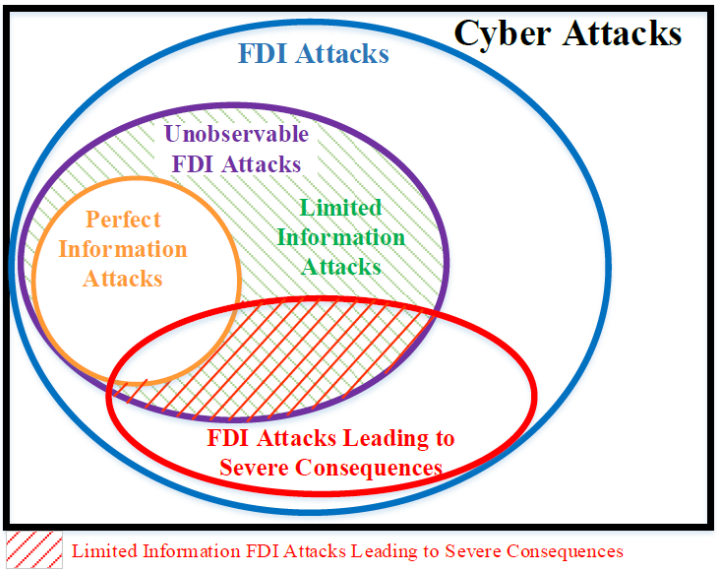


Figure 19. The space of cyber-attacks

4.2 Assumptions on attacker's knowledge and capability

We assume the attacker only has access to information inside an attack sub-network $\mathcal{L}$ and absolutely no knowledge of the outside network $\mathcal{E}$. In order to overcome the limited information, we suppose that the attacker infiltrates the sub-network long before it executes its attack, so that it can observe the natural behavior of the system in order to predict the effect of an attack. In particular, we assume that the attacker has access to historical data inside the sub-network that includes loads, costs, capacities, status, and dispatches of generators, and locational marginal prices (LMP). Historical data is sometimes directly utilized as pseudo-measurements to SE when the real-time information is incomplete. However, the attacker can be more sophisticated and use historical data to create higher fidelity boundary pseudo-measurements when they only have limited information. In this work, we suppose that the attacker uses multiple linear regression method to learn the relationship between the external network and the attack sub-network from

historical data. Furthermore, we predict the response of the control center under such attacks in a local sub-network via a bi-level optimization problem.

4.3 Worst-case line overflow attacks with localized information

4.3.1 System power flow with localized information

The attacker cannot calculate the line power flow inside $\mathcal{L}$ since both the PTDF matrix K of the entire network $\mathcal{G}$ and the subset of power injections in external network $\mathcal{E}$ are unavailable to attacker. To form the line power flow with limited information, we introduce a vector of pseudo-boundary injection $\bar{P}_{I,B}$. The ith entry of $\bar{P}_{I,B}$, namely $\bar{P}_{I,i}$, corresponding to boundary bus i, represents the sum of power flows delivered from $\mathcal{L}$ to $\mathcal{E}$ at boundary bus i, as

$$\bar{P}_{I,i} = \sum_{k \epsilon W_i^E} P_k$$

where W_i^E represents the lines located in $\mathcal{E}$ that are connected to boundary bus i. The vector of line power flows in $\mathcal{L}$ can then be written as

$$\bar{P} = \bar{K}(\bar{G}\bar{P}_G - \bar{P}_D) - \bar{K}^B\bar{P}_{I,B}$$

where $\bar{G}$ is the local generation to bus connectivity matrix, $\bar{P}_G$ is the local generation dispatch vector, $\bar{P}_D$ is the local loads, and $\bar{K}^B$ is the submatrix of K corresponding to boundary buses.

4.3.2 Multiple linear regression to predict pseudo-boundary injections

The optimal line overflow attack involves determining the attack vector in the first level and estimating the system response to the attack via the whole system DC OPF in the second level. However, due to limited knowledge, the attacker must predict the response of the OPF using only local knowledge. The OPF may be reformulated to include power balance, thermal limit, and generation limit constraints only in $\mathcal{L}$, and capture all effects in the external network through the pseudo-boundary injections $\bar{P}_{I,B}$. However, with this formulation, the attacker still cannot predict how the attack affects $\bar{P}_{I,B}$ since it depends on both power injections in $\mathcal{L}$ and $\mathcal{E}$. Therefore, before the attack is executed, the attacker cannot estimate the system re-dispatch after the attack accurately.

If the attacker can obtain a large amount of historical power injections and pseudo-boundary injections data in $\mathcal{L}$ (for example, by observing the system over a long time), it can learn a functional relationship between pseudo-boundary injection, $\bar{P}_{I,B}$, and power injections inside $\mathcal{L}$. The attacker can then predict the pseudo-boundary injections with the power injection in $\mathcal{L}$ as

$$\hat{\bar{P}}_{I,B} = F(\bar{G}\bar{P}_G - \bar{P}_D) + f$$

where $[F, f]$ represent an affine relationship, and $\hat{\bar{P}}_{I,B}$ is the attacker's prediction of pseudo-boundary injection by capturing the functional relationship via a linear model. Multiple linear regression is a statistical method to find a linear relationship between multiple inputs and single output, and it can be applied to predict the affine relationship $[F, f]$.

4.3.3 Attack optimization problem under localized information

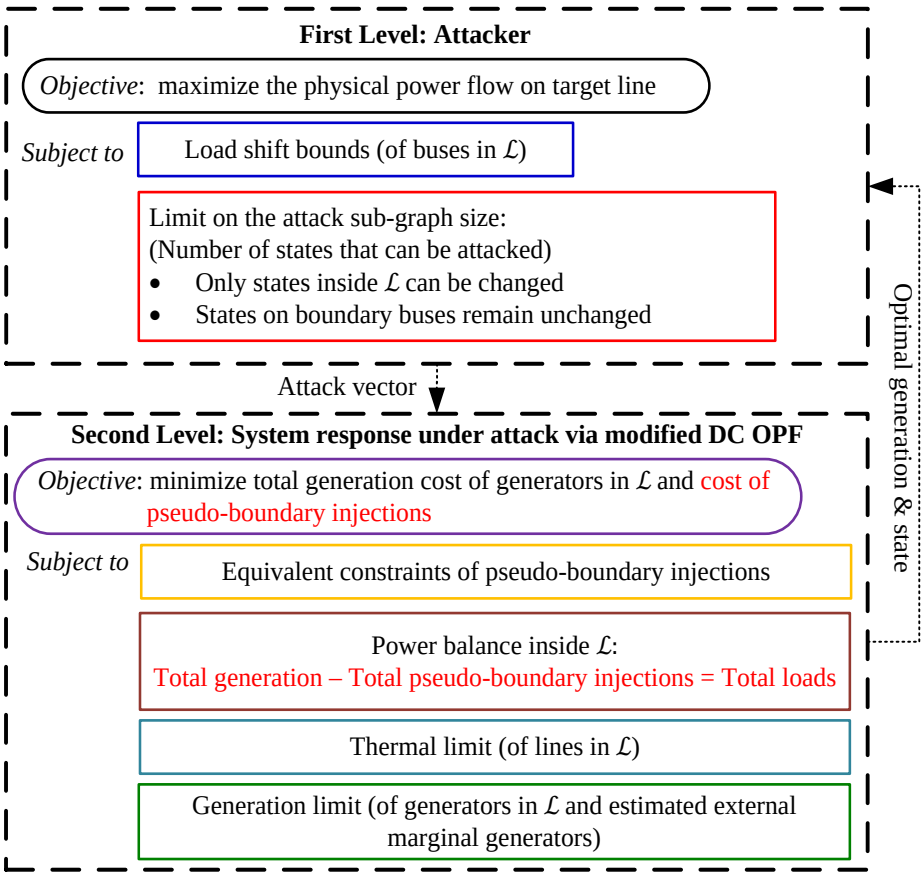


Figure 20. Bilevel optimization problem under limited information

Figure 20 illustrates the formulation of the bi-level attack optimization problem with limited information. The first level determines the attack vector in $\mathcal{L}$ that maximize target line flow and the second level represents system re-dispatch after attack via DC OPF formulated with only information in $\mathcal{L}$. However, since the attacker does not have knowledge of either the topology or the generator information in $\mathcal{E}$, we assume that the attacker only minimizes the total cost of generation in $\mathcal{L}$ and approximates the effect of the total generation cost in $\mathcal{E}$ as the total cost of the pseudo-boundary injections in the second level modified OPF. For boundary bus i, this cost is estimated as the product of the LMP, λ_i, and the pseudo-boundary injection at bus i. The equivalent constraints of pseudo-boundary injections is $\hat{\bar{P}}_{I,B} = F(\bar{G}\bar{P}_G - \bar{P}_D) + f$.

As with the original attack optimization problem for perfect information (as illustrated in Figure 13), this problem with limited information is non-linear and non-convex. We employ the same modifications as in [2] to convert it into a MILP.

Note that attacker can only overload lines in $\mathcal{L}$. The attack optimization problem ensures that only measurements inside $\mathcal{L}$ can be changed by attacker. The post-attack system re-dispatch (OPF), on the other side, forces all the cyber line power flows within the line limits. Therefore, the attacker can only hide physical overflows inside $\mathcal{L}$ with FDI attacks.

4.4 Simulation results

4.4.1 Simulation methodology

In this section, we illustrate the efficacy of the attacks designed with the method proposed in Sec. 4. To this end, we first compute the coefficient matrix with historical data using the multiple linear regression method. Subsequently, we solve the optimization problem to find the optimal attack vector inside $\mathcal{L}$. Finally, we test the physical consequences of the attack vector on the entire network $\mathcal{G}$. The test systems include the IEEE 24-bus reliability test system (RTS) and the IEEE 118-bus system from MATPOWER v4.1. In particular, the line rating data for IEEE 118-bus system is adopted from [29]. The whole network DC OPF and limited information attack algorithm is implemented with Matlab. The optimization problem is solved with CPLEX. The load shift is set to be 10%.

We focus on two scenarios for the historical data:

- **Scenario 1** - Constant Loads in $\mathcal{E}$: In each instance of data, loads in $\mathcal{E}$ remain unchanged while loads in $\mathcal{L}$ varies as a percent p of the base load, where p is independent $\mathcal{N}(0; 10\%)$. That is, power injections vary only at buses with marginal generators (denoted EM). The number of buses in EM is denoted by n_{EM}.

- **Scenario 2** - Varying Loads in the entire network $\mathcal{G}$: In each instance of data, loads in both $\mathcal{L}$ and $\mathcal{E}$ vary as a percent p of the base load, with p chosen independently for each load as $\mathcal{N}(0; 10\%)$. In this scenario, power injections at all buses in $\mathcal{E}$ vary in the historical data.

Note that the data in both scenarios also satisfy the following assumptions: (i) the topology for all the historical data remains the same, (ii) the historical generation dispatches data in both scenarios satisfies OPF, and (iii) there exists a subset of buses $\mathcal{Z}$ in $\mathcal{E}$, for which power injections remain constant in the historical data.

4.4.2 Results for IEEE 24-Bus RTS system

In this subsection, we present attack consequences on the IEEE 24-bus RTS system for Scenarios 1 and 2. The subnetwork L is illustrated in Figure 21. In each scenario, we compare the attack consequences on target line 28 determined by the optimization problems for two cases: (i) complete system knowledge (identified as global case), and (ii) limited system knowledge (henceforth identified as local case). For local case, we compare the physical power flow P_l^p and the attacker-computed physical power flow $\overline{P_l^p}$. The results of attacks are illustrated in Figure 22. We illustrate the difference between the physical and the attacker-computed pseudo-boundary injections in Figure 23.

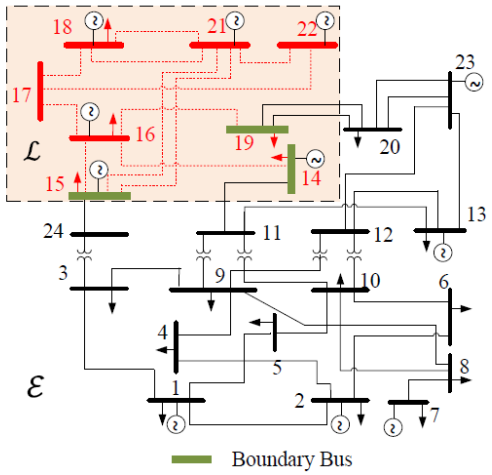


Figure 21. IEEE 24-bus RTS system decomposed into attack sub-network and external network

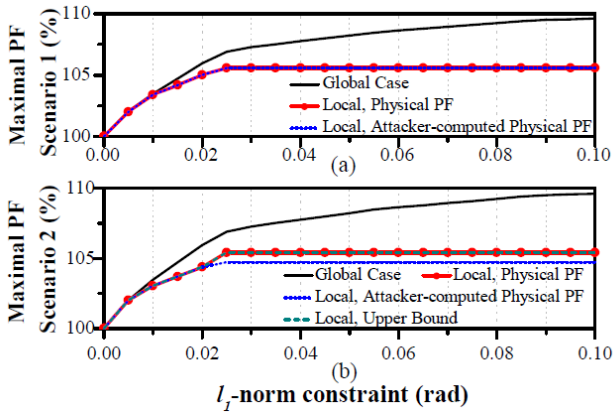


Figure 22. The maximum power flow (PF) v.s. the l_1-norm constraint (N1) when target line is 28 of IEEE 24-bus system for (a) Scenario 1, and (b) Scenario 2 historical data.

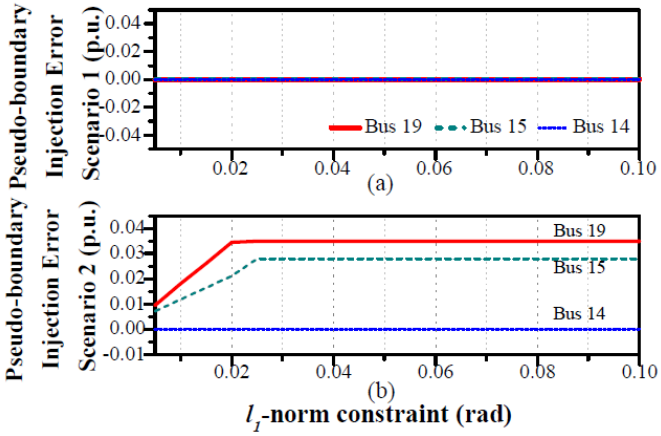


Figure 23. The pseudo-boundary power injection error v.s. the l_1-norm constraint (N1) when target line is 28 of IEEE 24-bus system for (a) Scenario 1, and (b) Scenario 2 historical data.

4.4.3 Results for IEEE 118-Bus system

The details of sub-network $\mathcal{L}$ are listed in Table 4. The results of attacks designed with historical data in Scenarios 1 and 2 are illustrated in Figure 24 with sub-plots (a) and (b), respectively. The difference between the physical and the attacker-computed pseudo-boundary injections at 3 of 18 boundary buses, buses 23, 70, and 80, for both scenarios are illustrated in Figure 25 with sub-plots (a) and (b), respectively.

Table 4 Summary of the attack sub-network in IEEE 118-bus system

Buses	1-14, 16, 17, 23, 25-27, 30, 33-35, 37-40, 47, 49, 59-66, 68-70, 75, 77, 80, 81, 116, 117
Lines	1-17, 20, 22, 31-33, 36-38, 47, 48, 50-55, 65, 88-100, 102, 104-108, 115, 116, 119, 120, 123, 124, 126, 127, 183, 184
Boundary buses	13, 14, 17, 23, 27, 33-35, 40, 47, 49, 59, 62, 66, 70, 75, 77, 80

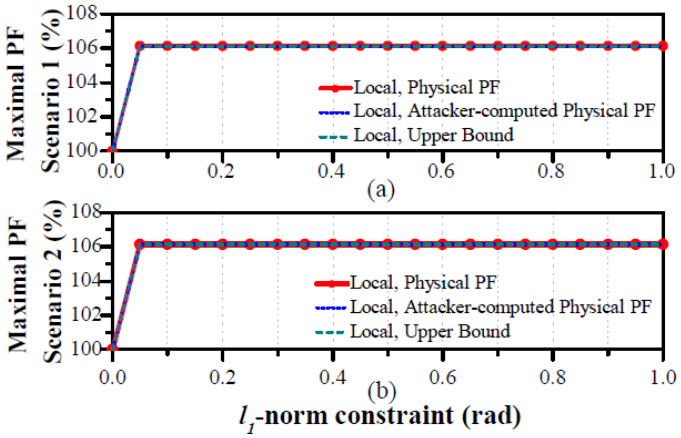


Figure 24. The maximum power flow (PF) v.s. the l_1-norm constraint (N1) when target line is 5 of IEEE 118-bus system for (a) Scenario 1, and (b) Scenario 2 historical data.

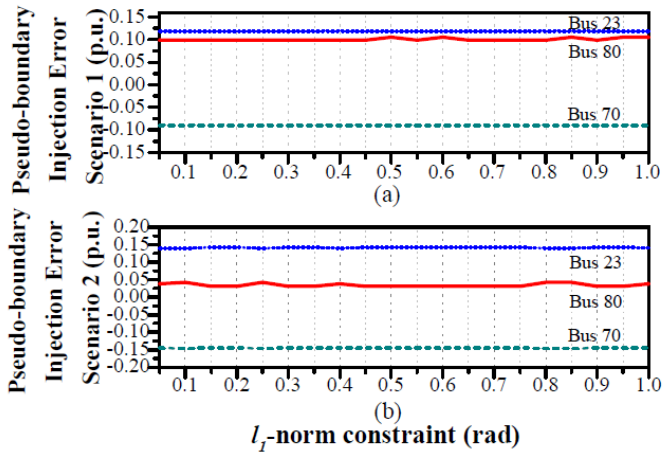


Figure 25. The pseudo-boundary power injection error v.s. the l_1-norm constraint (N1) when target line is 5 of IEEE 118-bus system for (a) Scenario 1, and (b) Scenario 2 historical data.

Figure 22 through Figure 24 demonstrates that even though there are mismatches between physical and attacker-computed pseudo-boundary injections, the attacker-computed physical power flow can still be correct. Note that, in this case, both the cyber power flow and the attacker-computed cyber power flow reach the limit post-attack since the target line is congested before attack. Therefore, the attacker-computed physical power flow is the same as the physical power flow.

4.4.4 Attack sensitivity to topology change

In this subsection, we evaluate the efficacy of attacks generated from historical data with topologies that are different from the real-time topology. We assume that the attacker uses the historical datasets to compute the coefficient matrices, and is not aware of a line outage in $\mathcal{E}$ in real-time. We exhaustively test the consequences of the attacks designed with the computed coefficient matrices on all possible real-time topologies with one line outage in $\mathcal{E}$. Note that the topology changes that will result in infeasible preattack DC OPF solution are not considered here. The l_1-norm constraints (N1) are chosen as 0.05 and 0.4 for the IEEE 24-bus and 118-bus systems, respectively. We compare the results with those in Secs. 4.4.2 and 4.4.3 and summarize them in Table 5. Note that for the pseudo-boundary power injection errors, we compare the l_2-norm of the errors on all boundary buses for each test case since there are multiple boundary buses in the test system. From the table, we can observe that if the attacker uses the coefficient matrix computed from historical data with different topology to design attacks, its evaluation of attack consequences may be undermined. Specifically for the case with Scenario 1 historical data in the IEEE 24-bus system, the attacker cannot obtain perfect prediction on pseudo-boundary injections any more since the real-time topology differs from that in the historical data. However, for most of the test cases, i.e., 86.47% and 98.26% cases for the IEEE 24-bus and 118-bus systems, respectively, the attacker can still cause line overflow with the inaccurate coefficient matrices.

Table 5. Summary of the sensitivity analysis results under different topologies

Test System & Scenario	# of Total Test Cases	% of Cases with Physical Overflow Decreases	% of Cases without Physical Overflow	% of Cases with Prediction Error Increases
24-bus SC1	22	18.18%	13.63%	100%
24-bus SC2	22	9.1%	13.63%	90.91%
118-bus SC1	115	0	0	63.48%
118-bus SC2	115	0	1.74%	10.43%

4.4.5 Verification of AC power flow model

In this subsection, we test the performance of the proposed attack strategies on AC power flow model. We first compare the attack consequences of the DC attacks in Secs. 4.4.2 and 4.4.3and the corresponding AC attacks in Figure 26. The AC attacks are computed with the designed DC attack vector. The system re-dispatch in response to each AC attack is via AC OPF. This figure validates that although the attack vector is solved by a linear optimization problem, it can still cause overflows in the AC system and the AC attack consequences track those of the original DC attacks. In addition, the impact of AC power flow historical data on the attack consequences are studied. We randomly generate a historical dataset, in which each instance is based on AC power flow model and satisfies all assumptions in Scenario 2. That is, the topology in each instance of

historical data is the same with the real-time topology. We then compute the coefficient matrix, solve the optimization problem to find the optimal attack, and test the physical consequences of the attack. The l_1-norm constraints (N1) are chosen as 0.05 and 0.4 for the IEEE 24- bus and 118-bus systems, respectively. We repeat this process 100 times and illustrate the results in Table 6. From this table, it can be seen that historical datasets with AC power flow data can reduce the prediction accuracy of the pseudo-boundary injections and the target line physical power flow. However, for both test systems, 100% of the designed attacks can result in physical target line overflows. These results demonstrate the robustness of the proposed attack strategy on AC power flow model.

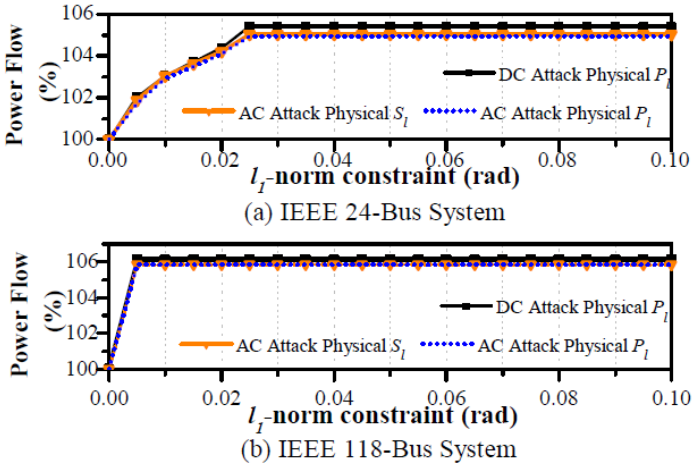


Figure 26. Comparison of the maximum power flow of DC and AC attacks.

Table 6. Summary of the sensitivity analysis results under AC power flow historical datasets

Test System	% of Cases with Physical Overflow	% of Cases without Physical Overflow Decreases	% of Cases with Prediction Error Increases
24-bus	100%	93%	46%
118-bus	100%	0	100%

5. Cyber-physical attacks

5.1 Motivation

Network topology is important system data used in various data processing modules in the EMS. Changes in topology can result from either system incidents or malicious physical attacks; but, in general, such topology alterations can be detected in the cyber layer. However, a sophisticated attacker can launch cyber-attacks that alter the topology information in an unobservable manner; furthermore, they can also mask a physical attack via a cyber-attack to create a more coordinated attack. Such cyber-physical attacks can result in wrong EMS solutions with potential serious consequences. Therefore, it is instructive to fully understand such attack consequences as a first step to thwart them.

Unobservable cyber-attacks on topology can be of two types: line-maintaining and line-removing. For a line-maintaining attack, the attacker changes measurements and line status information to make it appear that line that is not in the system is now shown as active at the control center via SCADA data; the opposite is achieved by a line-removing attack. For both line-removing and line- maintaining attacks, an attack can either change only topology data (i.e., state-preserving topology attack) or both state and topology data (i.e., state-and-topology attack). The class of unobservable cyber topology attacks is first introduced in [30]; however, the analysis in [30] is restricted to a subclass of state-preserving line-removing attacks in which an attacker changes topology information of the system without changing the states.

We focus on line-maintaining attacks that requires both physical line outage and cyber-attack to mask the physical topology alteration, *i.e.*, both physical and cyber topology are changed by the attacker. In [31], unobservable state-preserving line-maintaining attacks (i.e., only topology data is changed) are studied. However, changing only topology and not changing states limits the number of feasible lines amenable to attack and also requires large load shifts at the end buses of a target line. Therefore, we determine attacks that change both state and topology.

5.1 System and attack model

5.1.1 System model

The electric power system can be represented by a graph G. The state estimation is then given by

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}, \mathrm{G}) + \boldsymbol{e}$$

where x is the system state vector, and e is a noise vector which is independent of x and is modeled as Gaussian distributed with 0 mean. The function $\boldsymbol{h}(\boldsymbol{x}, \mathrm{G})$ is a vector of nonlinear functions that describes the relationship between the system states and measurements for a topology G. In Figure *27*, we illustrate a typical temporal sequence of data processing units in the cyber layer.

5.1.2 Attack model

The unobservable state-and-topology cyber-physical attack considered here models both a physical attack and a coordinated cyber-attack.

We assume the attacker has the following capabilities:

1. Attacker has knowledge of the topology G_0 of entire network prior to physical attacks.
2. Attacker has the capability to launch physical attack and observe and change measurements only for a sub-graph S of G_0. The choice of S is described in detail in the sequel.
3. Attacker has the capability to perform SE and compute modified measurements for S.
4. Attacker has knowledge of the capacity and operation cost of every generator in the network.
5. Attacker has historic data of load patterns and generation dispatch of the entire network.

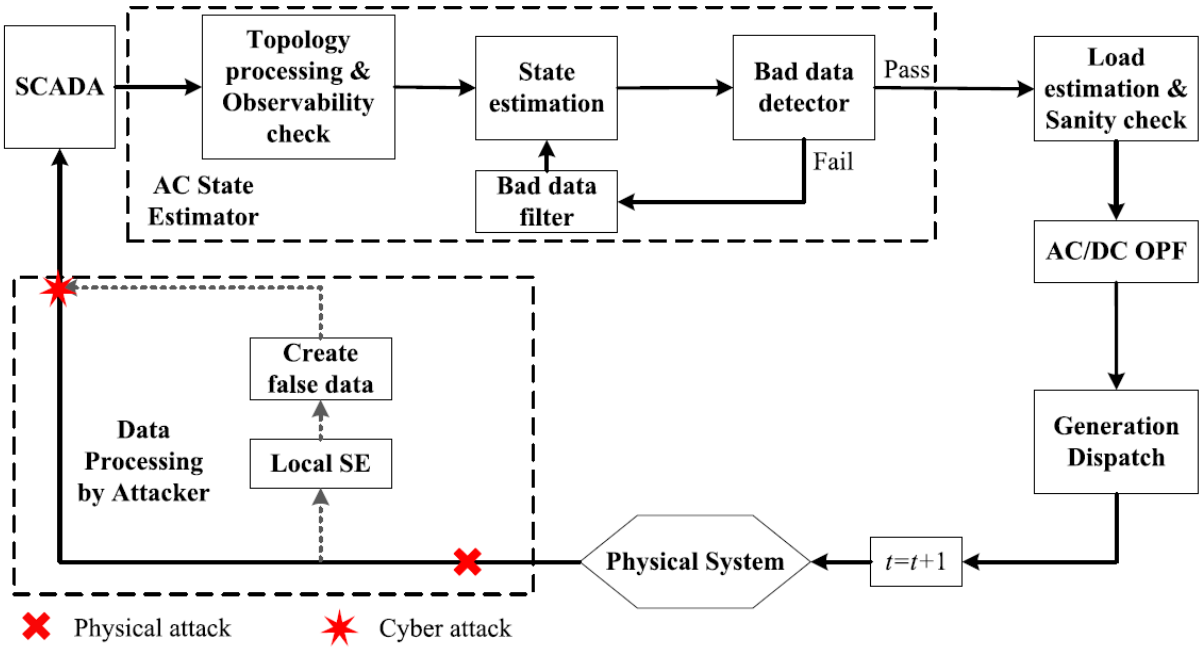


Figure 27 Temporal Sequence of Data Processing Units in The Cyber Layer within Attack

We assume that the power system is observable before and after the physical attack. We assume that an intelligent attacker can learn the system information a priori, e.g., by hacking into the system databases and learning the system models and functions ahead of time. Moreover, attacker can also replace the generation data with public market data such as locational marginal prices and quantities to construct the attacks.

We denote the line that is physically tripped by the attacker as the switching attack line and the two end buses of this line as the switching attack buses. Assume the switching attack line is line *t* and the topology prior to the physical attack is G0. The physical line status for line *t* changes from s_t= 1 to s_t= 0 after the physical attack and the corresponding physical topology changes to G.

In general, a physical attack will be subsequently detected by the topology processing unit in the EMS and the system topology will be updated shortly after the detection. However, a sophisticated attacker can hide such physical attacks by launching an unobservable cyber-attack. In the resulting unobservable cyber topology attack, the attacker modifies line status as well as related bus measurements to alter the system topology G to a different "target" topology $\bar{G}$. Since the attacker's aim is to hide the topology alteration caused by the physical attack, $\bar{G}$ should be chosen as G0.

To launch a state-and-topology attack, the attacker injects line status attack vector b and measurement attack vector a. The attack vector b for line status overrides the physical change on line t's status by setting for b_k=0 for $k \neq t$ and b_k=1 for $k = t$. These changes lead to a new system state $\bar{x}$ for the system under attack. This attack modifies (s, z) for topology G to $(\bar{s}, \bar{z})$ for topology $\bar{G}$ such that

$$\bar{s} = s + b, \quad \bar{z} = z + a.$$

In the absence of noise, the measurement attack vector satisfies

$$a = h(\bar{x}, \bar{G}) - h(x, G).$$

For nonlinear measurement model and AC SE, we model a sophisticated attacker who attacks measurements and line status data for a sub-graph S of the network by first estimating the system states $\hat{x}$ inside S using AC SE. The attacker then chooses a small set of buses in S to change states from the estimate $\hat{x}$ to $\bar{x} = \hat{x} + c$ such that the measurement vector $\bar{z}$ after cyber-attack has entries

$$\bar{z}_i = \begin{cases} z_i, & i \notin I_S \\ h_i(\hat{x} + c, \bar{G}), & i \in I_S \end{cases}$$

where I_S denotes the set of measurements inside S.

We use the following method to identify the sub-graph S for an unobservable state-and-topology attack. Throughout, we distinguish two types of buses: load buses with presence of load and non-load buses with no load.

1. Use the optimization problem (the details are in the sequel) to determine the load buses from the attack vector c whose states need to be changed (defined as center bus) to enable the attack.
2. Include all center buses in S.
3. Extend S by including all buses and branches connected to the buses inside S.
4. If there are non-load buses on the boundary of S, extend S by including all adjacent buses of the non-load boundary buses and the corresponding branches.
5. Repeat 4) until all boundary buses of S are load buses.
6. Check if there is a path (actual bus and branch connection) in S that can connect the two switching attack buses. If such path exits, then S is the attack sub-graph. If there is no such path, go to Step 7).

7. Use BFS method to find the shortest path connecting the two switching attack buses. Include the shortest path in S. Then this S is the attack sub-graph.

5.2 Attack strategy

In this section, we study the worst-case cyber-physical attacks with the following capabilities: (a) physically trip a switching attack line and mask the physical attack with a cyber-attack; (b) maximize power flow on a target line; and (c) avoid detectability by limiting load shift via changes in measurements. The attack resources available to the attacker may also be limited. We model this limitation by constraining the size of sub-network the attacker has access to. This leads to a constrained optimization problem. As noted before, two attack vectors are needed since both physical and cyber-attacks result in state and topology changes.

Our two-step optimization problem captures the temporal nature of attack sequence involving a physical attack followed by several cyber-attacks. In Figure 28, we illustrate this temporal sequence of attack and system events. The system events are periodic and are denoted by S_t for the t_th event. At the start of each S_t, data is collected from SCADA and by the end of S_t, i.e., the start of S_(t+1), data is processed in the EMS. There are 2 attack instances, A_0 and A_1 to denote the physical and cyber-attack events, respectively. We assume the physical attack event A_0 is launched immediately after the start of the 0_th system event, i.e., S_0, and the coordinated cyber-attack event A_1 is launched shortly after, but before the start of next system event S_1. Following this cyber-physical attack pair (A_0, A_1), the cyber-attack is sustained between every two system events to maintain the worst generation dispatch, and thereby, sustain the maximal power flow on the target line. In Table 7, we denote how the cyber (measured) and physical (actual) data including generation dispatch, system state, topology, and loads vary at all system and attack events.

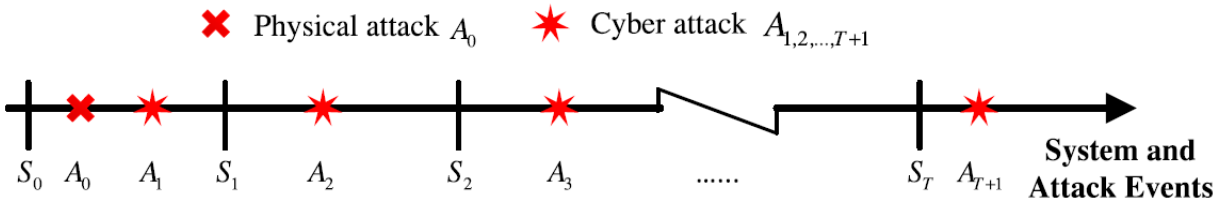


Figure 28 Time sequence of attack and system events.

Table 7 Physical and cyber data for attack and system events

Event	S_0	A_0	A_1	S_1	A_2	S_2	A_3	…	S_T	S_{T+1}
Generation dispatch	P_G^0	P_G^0	P_G^0	P_G^*	P_G^*	P_G^*	P_G^*	…	P_G^*	P_G^*
Physical topology	$\bar{G}$	G	G	G	G	G	G	…	G	G
Cyber topology	$\bar{G}$	G	$\bar{G}$	$\bar{G}$	$\bar{G}$	$\bar{G}$	$\bar{G}$	…	$\bar{G}$	$\bar{G}$
Physical state	θ_{0-}	θ_0	θ_0	θ^*	θ^*	θ^*	θ^*	…	θ^*	θ^*
Cyber state	θ_{0-}	θ_0	$\theta_0 + c^0$	$\theta^* + c$	$\theta^* + c$	$\theta^* + c$	$\theta^* + c$	…	$\theta^* + c$	$\theta^* + c$
Physical load	P_D	P_D	P_D	P_D	P_D	P_D	P_D	…	P_D	P_D
Cyber load	P_D	P_D	$\bar{P}_D$	$\bar{P}_D$	$\bar{P}_D$	$\bar{P}_D$	$\bar{P}_D$	…	$\bar{P}_D$	$\bar{P}_D$

Assume the system topology and the generation at S_0 are $\bar{G}$ and P_G^0, respectively. From Table I, we can see that the system physical topology changes to G after the physical attack in A_0. The physical operation states, thereby, change to θ_0. The attacker then injects cyber-attack vector c^0 in A_1 to change the load pattern from the physical load P_D to the false cyber load $\bar{P}_D$ to mask the physical topology alteration.

The physical and cyber loads at attack event A1 satisfy the following relationship:

$$\bar{P}_D = P_D + H_1\theta_0 - \bar{H}_1(\theta_0 + c^0)$$

where H_1 and $\bar{H}_1$ are dependency matrices between power injection and voltage angle for G and $\bar{G}$, respectively. The false cyber load $\bar{P}_D$ and topology $\bar{G}$ leads to a system redispatch to the optimal generation dispatch P_G^* at S_1. Since the attacker optimization problem at each step models the system response, such an optimal dispatch will cause maximal power flow on the target line. Following this first cyber-attack A_1, since the generation dispatch changes at S_1, the physical system states also change to θ^*. To sustain both the optimal dispatch P_G^* and the false cyber topology $\bar{G}$ at the next system event, i.e., S_2, the attacker needs to maintain the false cyber load by injecting another attack vector c at A_2. In the following attack events, the attacker can keep injecting c to maintain the false cyber load. This in turn ensures that the optimal dispatch and the false cyber topology are maintained and the maximal power flow on the target line is sustained.

To model the cyber-physical attack events A_0, A_1, and A_2 between S_0 and S_1, the optimization problem should capture the power balance relationship. However, since the switching attack line is determined by the optimization problem, both H_1 and θ_0 are unknown before solving the problem. On the other hand, for the pure cyber-attack events A_2 and A_3, the power balance in the cyber layer is equivalent to the physical power balance. For each bus, the power injection also equals to the sum of power flow on the branches connecting to the bus. Therefore, we can use the

vector of the sum of power flow on the branches connecting to each bus, to replace $H_1\theta$ and eliminate H_1.

Therefore, instead of directly modeling the physical attack event A_0 and cyber-attack events A_1 and A_2 between S_0 and S_1, we use the first step optimization problem to model the pure cyber-attack events A_2 and A_3 between S_1 and S_2 to determine the attack vector c for a fixed but undermined topology G. Such a c should be subject to bounds on both the attacker's sub-graph size and the load shifts and can lead to the worst dispatch P_G^*. Note that P_G^* is the worst dispatch that can lead to the maximal power flow on the target line under physical topology G and physical load. Once the line to attack is determined, we need the second step optimization to find the attack vector c^0 that also ensures the worst dispatch P_G^*.

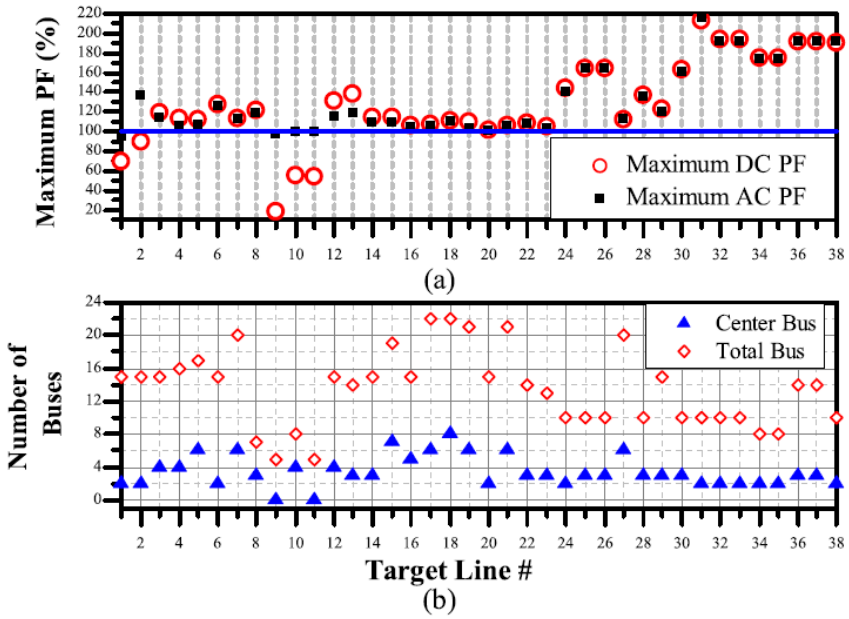
In the second step of the optimization, we compute the attack vector c^0 at A_1. We again use a two-stage optimization problem to determine the c^0 such that the optimal generation dispatch for this problem is forced to be same as P_G^*. We, henceforth, define the attack vector solved in the second step as the initial attack vector. This step can be assumed as an on-line attack determination since it requires the real-time physical states data.

5.3 Numerical results

In this section, we test the effect of attacks designed with the two-step attack strategy for a nonlinear system model. The test system is the IEEE 24-bus reliable test system (RTS). We assume: (i) the system is operating under optimal power flow; and (ii) the loads of the system are constant and are equivalent to the historic load data that is assumed to be known to the attacker. To model realistic power systems, we assume that there are congestions prior to the attack and the attacker chooses one congested line as target to maximize power flow. We use MATPOWER to run AC power flow and AC OPF. The optimization problem is solved with CPLEX.

5.3.1 Solution for the attack designed with the attack strategy

In order to understand the worst-case effect of attacks, we assume there is a line congested prior to the physical attack. This is achieved in simulation by reducing the line rating to 95% of the base case power flow (apparent power) to create congestion. We exhaustively test all 38 lines as targets in the system. Figure 29 illustrates the maximal power flow (PF) and attack size (# of buses in sub-graph) for load shift bounds τ = 10%, total lines to physically attack N_T = 1, and the l_1-norm constraint N_1= 0.06. From Figure 29(a) we can observe that the attack vector determined by the two-stage optimization problem cause overflows in 33 target lines in linear model, i.e., 86.84% of the attacks are successful. For all such successful attacks, using the attack vector to construct an attack in the nonlinear model, in Figure 29(a), the AC PF in each line tracks DC PF solved with the attack strategy. In particular, 2 cases with target lines 9 and 11, respectively, have no center buses, i.e., for these lines the state-preserving attacks introduced in [31] suffice. In Figure 29(b), we can observe that 72.73% of the successful attacks can be launched inside a sub-graph with less than 16 total buses.



(a) Maximum DC & AC PF (b) Number of Center Buses & Total Buses inside Attack Sub-graph v.s. Target Line.

Figure 29 *Summary of all 38 target lines under attack*

5.3.2 Consequences of the Attack in the Nonlinear Model

In this subsection, we select a typical case to demonstrate the consequence of the unobservable state-and-topology cyber-physical attack determined by the attack strategy in the nonlinear system model. In this case, the target line is line 12. Under this condition, the switching attack line is line 2.

For the chosen target line, after launching the physical attack at A_0 and injecting the initial cyber-attack constructed with c^0 at A_0, the active power generation dispatch for generators at bus 7 and 13 change from 215.69 MW and 230.96 MW to 200.69 MW and 245.67 MW, respectively (the dispatch of other generators remains unchanged). In the following events, as the attacker continues to inject the AC attacks constructed with attack vector *c* (determined by Step 1 optimization), the active power generation for these two set of generators are maintained at these values. Figure 30 demonstrates the cyber and physical power flow variation during 20 system events. From Figure 30, we can observe that once the active power generation dispatch changes to the optimal dispatch and remains unchanged in the subsequent system events, the physical overflow in the target line will be maintained by injecting the AC attack constructed with attack vector c. The heat accumulation may eventually cause this line to overheat and then trip offline all the while remaining unobservable to the control center.

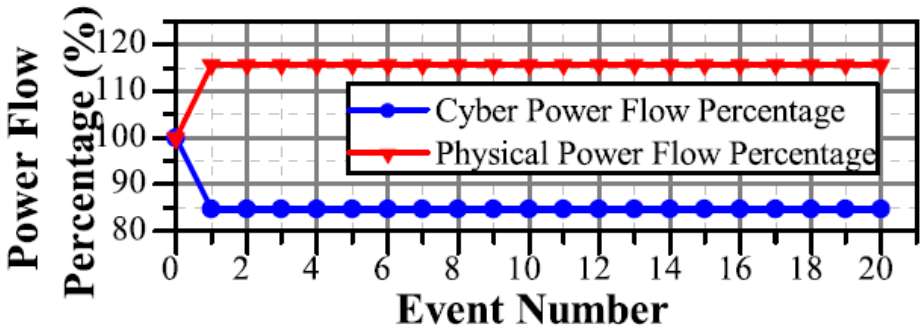


Figure 30 Power flow variation on line 12 during 20 system events

6. Countermeasures

The load redistribution attacks described in the previous sections have been proven and shown to be unobservable to current state-of-the-art bad data detectors in EMSs. For this reason we now develop countermeasures against this wide class of attacks in the form of detectors that analyze the measured loads, before they are used to perform economic dispatch, and determine if they represent a realistic state of the system or if they have been maliciously engineered. Our approach exploits the fact that system operators have access to historical load data which can be used to learn patterns in an offline manner and design machine learning algorithms that can check load consistency in real time scenarios. We use nearest neighbor algorithm, support vector machines and replicator neural networks to design three different detectors which are tested on realistic historical load data created for the IEEE 30-bus system. To design a detector that works for any possible load pattern that might occur, we exploit publicly available PJM zonal data [32] to create realistic load data for multiple years for the IEEE 30-bus system.

Other attempts have been made to design detection algorithms and countermeasures against cyber-attacks. In [33] for example, sparse optimization is leveraged to detect corrupted measurements while in [34] the detection is performed by using phasor measurements. In these studies, SE is used as a tool for identifying attacked measurements without any considerations on the system state they represent. Our work differs in that by applying machine learning on historical data we are able to check for consistency of the observed load data and judge if it corresponds to a realistic system configuration. In [35], machine learning techniques are applied to SE for anomaly detection but the models used to test the detection algorithms are not representative of the behavior of a real system because only the base case operating point is considered. The training data used by the authors in [35] represents measurements of the same system configuration and differs only for the random noise that is added. This makes it possible to find very efficient detection schemes, which are not guaranteed to perform as well when the system loads are different from the base case.

6.1 System model

6.1.1 Load data: model and design

Throughout our analysis we use the IEEE 30-bus system and real load data published by PJM. On its website, PJM regularly provides the hourly loads of the 20 zones of the northeastern US grid. This historical data is available starting from the year 1993 and it is updated monthly with the most recent figures. We leverage the fact that the 30-bus system contains 20 load buses to create a test case with realistic load profiles, by mapping each PJM zonal load to a corresponding load in the IEEE system. The zonal loads and the loads in the IEEE 30-bus system are ranked by relative size compared to the total load size of their respective system, as shown in Table 8.

Table 8. Relative size of PJM zones and 30-bus system loads

PJM zone		30-bus system load	
Zone name	**Size [%]**	**Bus location**	**size [%]**
AEP	14.5	**8**	15.9
CE	13.7	**7**	12.1
DOM	12.4	**2**	11.5
ATSI	8.25	**21**	9.25
PS	6.34	**12**	5.92
AP	5.59	**30**	5.60
PE	5.41	**19**	5.02
PL	4.51	**17**	4.76
BC	4.27	**24**	4.60
PEP	4.08	**15**	4.33
JC	3.85	**4**	4.02
DEOK	3.35	**14**	3.28
DPL	2.60	**10**	3.07
DAY	2.14	**16**	1.85
ME	1.89	**26**	1.85
PN	1.88	**18**	1.69
DUQ	1.81	**23**	1.69
AE	1.73	**3**	1.27
EKPC	1.46	**29**	1.27
RECO	0.26	**20**	1.16

The percentage distribution of the loads of the PJM system is very close to that of the loads in the 30-bus system, which justifies the mapping from each zone to the similarly sized load in our test system. The mapping has been performed by first determining a mapping ratio between the PJM and the 30 bus system and then multiplying every hourly load of the 20 zones by this factor. In this way, we created new load configurations corresponding to every hour of the year for our test system. Creating load configurations in which many lines are congested or at their rated limits maximizes the chances of finding successful attacks since a congested system often represents a worst-case scenario in terms of vulnerability to cyber-attacks and possible physical consequences. Thus, we defined the mapping ratio as the ratio between the net load of the 30-bus system base case and the maximum net load of the PJM system:

$$m_{ratio} = \frac{30\ bus\ net\ load}{max\ PJM\ net\ load} \times k = \frac{189.2\ MW}{144644\ MW} \times 1.39 = 1.308 \times 10^{-3}$$

where k is a constant that was chosen in order to obtain a system with moderate amount of congestion. We used the PJM data to create a load dataset for the IEEE 30-bus system for five

years, corresponding to the period from January 2012 to December 2016 which we assumed to be our *normal* data. In the detection experiments, the first four years are used as historical data to train the detectors and the last year is used as the test data. Thus, the training data contains 35064 distinct load profiles (one for each hour), while the test data comprises 8784 load profiles.

6.1.2 Attack model and design

In order to test the detection capability of the proposed algorithms, we design many attacks following the model presented in Section 3.2. After computing the attack vector, the set of false measurements is passed to the state estimator which calculates the corresponding loads. It is this set of attacked loads that need to be flagged by the proposed detectors as malicious. We use this attack strategy to compute attacks on every hour of the 2016 synthetic data during which one or more lines were congested. A DCOPF is run on each of the hourly load profiles to measure the level of congestion in the system. Every time a solution showed a line at or above 85% of its rating, an attack was computed. We define an attack as successful if it leads to a flow on the target line greater than 100% of its rating. The results of this process are summarized below:

DCOPF results

- 1197 hours out of 8784 total hours with at least one line above 85% rating;
- 450 hours out of 1197 hours with at least one line at 100%.

Successful attacks on congested hours

- LS = 10% : 437 successful attacks;
- LS = 15% : 479 successful attacks.
- As shown by these results, depending on the specific load configuration at each hour, the search for an attack is not always successful. Moreover, the greater the allowable load shift, the higher the chances of computing a successful attack.

6.2 Detection algorithms

We propose three detectors which analyze the measured loads of a power system during any hour of the day and determine if they are normal loads or if they have been maliciously modified through a cyber-attack. Each detector is based on a different machine learning technique, but the general approach in determining the soundness of a set of measurements is similar. The measured load configuration to be tested is given as input to the detector, which generates a scalar value based on a metric specific to the machine learning technique used. This value is compared against a predetermined threshold to label the loads as *normal* or *attacked*. The specific value of the threshold is chosen as a tradeoff between detection probability and false alarm rate.

6.2.1 Nearest neighbor algorithm

Nearest neighbor algorithms are based on the assumption that data labeled as normal lies in limited, dense regions of space while anomalies are located further from these neighborhoods [36], [37], see Figure 31. Define $\boldsymbol{l_i}$ as the $1 \times n_l$ vector of loads at time i, where n_l is the number of loads in

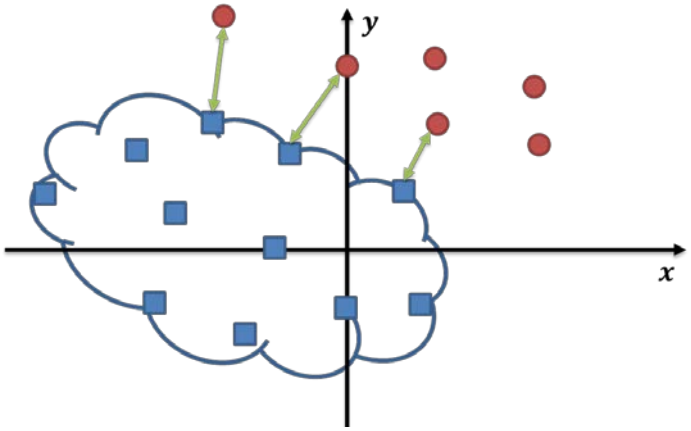


Figure 31. Graphical representation of the concept of nearest neighbor

the system. The normal data is represented by the n_h historical load vectors which have been measured in the past and we indicate as $\boldsymbol{h_j}$, for $j = [1{:}\, n_h]$. The classification is done by measuring the Euclidean distance between the current load profile $\boldsymbol{l_i}$ and every vector $\boldsymbol{h_j}$ in the historical data set (assumed to be attack free). The closest distance d_i for a sample $\boldsymbol{l_i}$ is defined as

$$d_i = \min_{j=[1:n_h]} \left\| \boldsymbol{l_i} - \boldsymbol{h_j} \right\|_2.$$

To label $\boldsymbol{l_i}$ as normal or attacked, d_i is compared against the predetermined threshold.

6.2.2 Support vector machine

The second machine learning technique we tested is a support vector machine (SVM), in which data is used to define a boundary of the region of space in which all the normal points lie [37]. The training of an SVM results in identifying a hyperplane which includes all of the normal training data, and none of the abnormal training data, as shown by Figure 32. For non-linear classification, the training phase uses kernels to map the data to high-dimensional spaces making it possible to learn complex regions. Training of SVMs can be supervised or semi-supervised: in the former, both *normal* and *abnormal* data is used for learning, while in the latter only *normal* instances are considered.

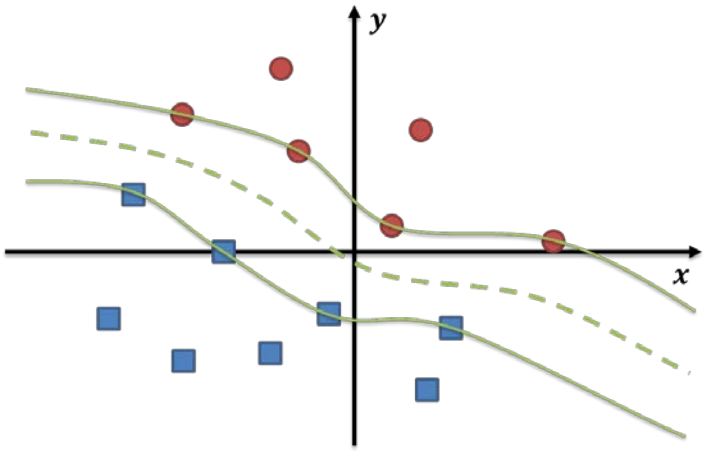


Figure 32. Graphical representation of the concept of support vector machine

In our tests, the data points used for training are the historical load vectors $\boldsymbol{h_j}$ from 2012 to 2015. All historical points are labeled as 1, indicating that they are normal loads, making this a semi-supervised machine learning algorithm. Preliminary testing showed that a linear hyperplane is not effective in classifying the load data, so instead we use a Gaussian radial basis kernel function to create a complex, nonlinear boundary [38].

The testing phase consists in feeding the normal load vectors for 2016 and the attacked load vectors to the SVM; for each load instance, the SVM computes a score between [-1,1]. The closer a score is to +1, the higher the confidence that the loads are normal; vice versa, scores close to -1 indicate that the loads have likely been maliciously modified by an attacker. These scores are the metric which is compared against a threshold to label the observed loads of the system as normal or attacked.

6.2.3 Replicator neural network

Replicator neural networks are a particular type of neural networks which are trained to compress and then reconstruct the data that is fed to them. For this reason, replicator neural networks have the same number of output neurons as the number of inputs. The neural network topology typically used includes three hidden layers with an equal number of hidden nodes each. The training phase aims at creating connections that are able to compress the input data (which usually has higher dimensionality than the number of nodes in each hidden layer) and then reconstruct it at the output nodes minimizing the error between input and output. The anomaly detection is performed by feeding to the trained network the data to be tested and measuring the discrepancy between input and output. For each load vector $\boldsymbol{l_i}$ at time i the error is computed as

$$\delta_i = \|\boldsymbol{l_i} - \boldsymbol{o_i}\|_2$$

where $\boldsymbol{o_i}$ is the output load vector of the neural network. Comparing the replication error with a set threshold allows for the labeling of the data point as normal or attacked.

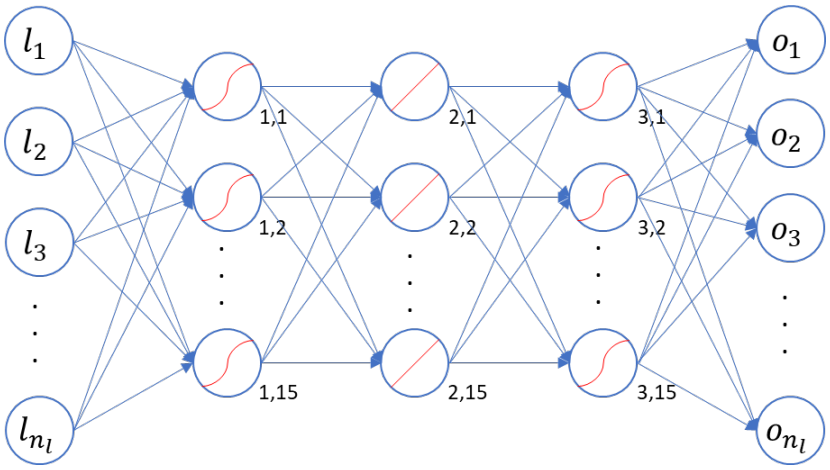


Figure 33. Model of the replicator neural network used

Intuitively, this neural network is trained to learn a model of the correlation between the different loads in the system. Real loads will have a close correspondence to the model learnt and they will be reproduced with a small error, while a sample with loads that have been maliciously modified will yield bigger replication errors. In this work we used a neural network, shown in Figure 33, made of three hidden layers, each consisting of 15 nodes. The nodes in layers 1 and 3 have a sigmoid activation function, while the nodes in layer 2 have a linear activation function. We also tested a similar neural network, with all layers having sigmoid activation functions but it did not perform as well.

The data used for training is the load profiles from 2012 to 2015. In the training process, each sample is fed to the network and the weights of the internal connections are adjusted to minimize the difference between input and output. This process is performed through backpropagation of the error and it is solved using Levenberg-Marquardt optimization with mean squared error as performance function [39], [40].

6.3 Experiments

6.3.1 Experimental methodology

The testing methodology we follow consists of two steps and is the same for each detector. First, we compute the detector specific metric (minimum distance, score, or replication error) for each load configuration in the test data of 2016 and then we evaluate the performance in terms of *missed detection* (M_D) and *false alarm rate* (F_A). Missed detection is defined as the ratio of the number of attacked cases flagged as normal to the total number of attacked cases, while false alarm rate is the ratio of the number of cases in which normal data is flagged as attacked to the total number of normal cases (8784 hours). These two metrics are evaluated for the dataset across a range of detection thresholds, thereby characterizing the tradeoff between missed detection and false alarm rate. These results are used to plot the receiver operating characteristic (ROC).

6.3.2 Experimental results

For the nearest neighbor algorithm, the minimum distance of every normal and attacked case from the closest sample in the training dataset is presented in Figure 34. For each hour of the year (x-axis) the minimum distance is plotted on the y-axis: the blue asterisks correspond to normal loads, while the green cross signs and red plus signs represent the attacked cases, with 10% and 15% load shift respectively. Similarly,

Figure 35 shows the scores computed using the SVM and Figure 36 shows the replication error of the neural network based detector. The performance of the detectors with 10% LS attacks is evaluated by plotting the ROC in Figure 37.

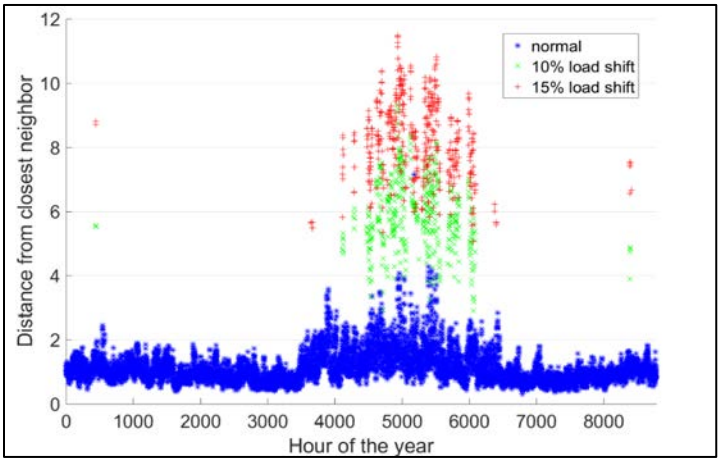


Figure 34. Distribution of nearest neighbor distance for normal and attacked cases

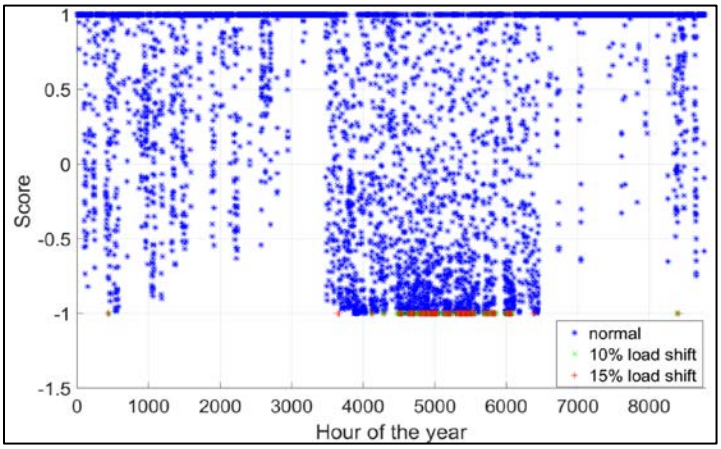


Figure 35. Distribution of SVM scores for normal and attacked cases. Note that for SVM a higher score (closer to 1) represents a belief that the data is normal, whereas a lower score (closer to -1) represents attacked data

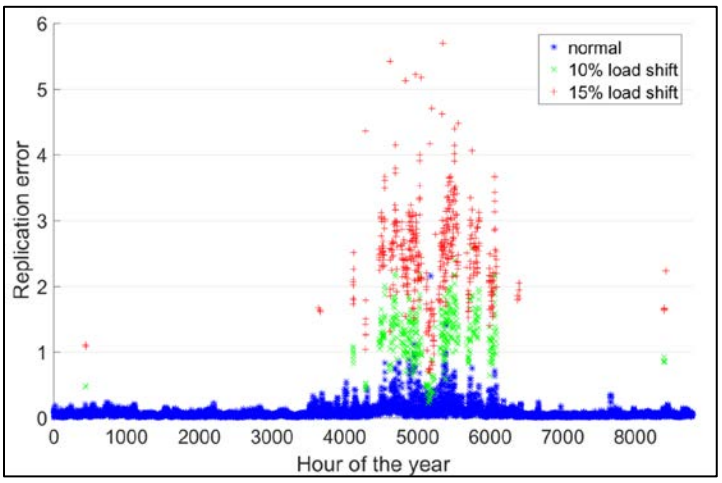


Figure 36. Distribution of replication error from the replicator neural network detector for normal and attacked cases

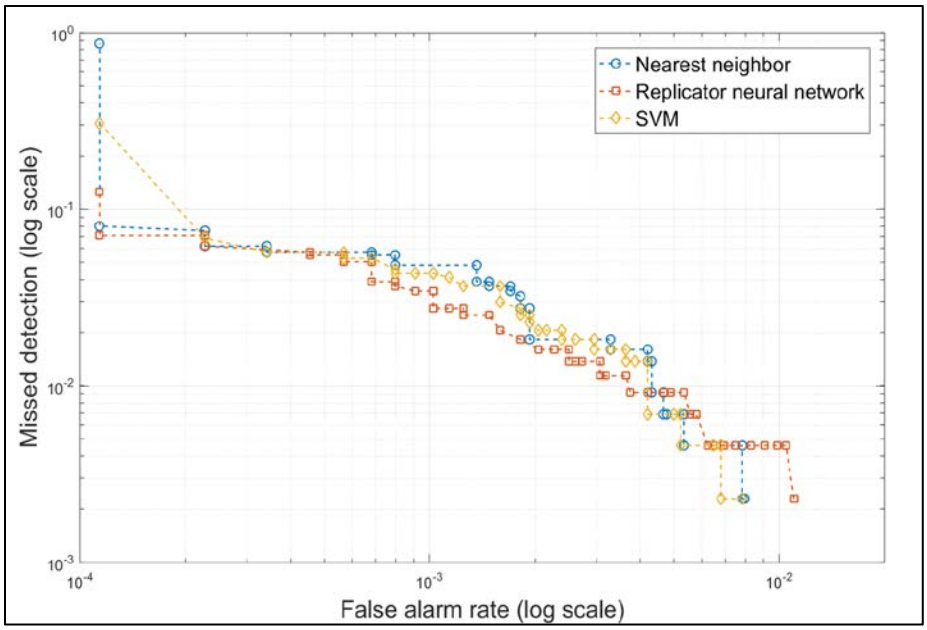


Figure 37. Receiver operating characteristic of the three detectors with 10% load shift attacks

In the case of 15% LS, detecting the attacks becomes trivial, as shown in Table 9. For each detector, the values of missed detection and false alarm rate are shown for two different detection thresholds highlighting the near perfect performance of the detectors. Overall the results of the three detectors are very similar. An important difference in the case of the neural network-based detectors is the computational complexity. The training phase for our neural network required almost 6 hours to complete using the Matlab machine learning toolbox, while the training of the SVM on the same data took less than 1 minute. The nearest neighbor algorithm does not necessitate a training phase, but unlike the other two methods all the computing is done in real time by searching for the

minimum distance. This search requires the calculation of the distance from every entry in the historical data, but in our tests this operation only takes a fraction of a second. In terms of applications to real time detection, after performing the training of SVM and replicator neural network offline, all three detectors are able to analyze the loads much faster than the sampling rate of modern SCADA systems.

Table 9. Performance of the detectors with 15% LS attacks

Detector	M_D	F_A	*# of false alarms*
Nearest neighbor	0.2192	0	0
	0	1.138×10^{-4}	1
SVM	0.0021	1.138×10^{-4}	1
	0	2.277×10^{-4}	2
Replicator	0.0125	0	0
Neural Network	0	1.138×10^{-4}	1

7. Concluding remarks

In this project, the risk of well-coordinated sophisticated cyber-attacks on the operation of power systems has been studied. The scalable tools we developed to analyze the vulnerabilities of large scale systems have been validated on a realistic EMS platform; system operators can now use these algorithms to identify criticalities in real world applications. Moreover, the countermeasures we designed in the form of detectors which recognize counterfeit data can easily be verified and then implemented in any EMS as they rely on data which is already being collected at the control centers. Overall, this research benefit ISOs, utilities, and vendors by providing a holistic analysis of credible and consequential cyber-threats and developing attack-resilient detection and control algorithms via a realistic software EMS simulation platform.

The grant that supported this project allowed us to demonstrate our work and our results to PSERC members as well as to the Department of Homeland Security and the National Science Foundation. There are a number of fundamental challenges that contribute to the ongoing debate as to how approach the problem of cyber-attacks against the electrical grid. For this particular project, it was critical to have a constant dialogue between academia and industry. The PIs attended meetings with the Industry Advisory Board in December 2016, May 2017, December 2017, and June 2018 where feedback from industry experts was gathered and used to improve our work. Moreover, at these meetings, the students presented the latest results with multiple posters and demoed the attacks and countermeasures on the EMS platform we developed. Finally, PI Sankar is working with researchers at GE to further improve this work and extend it to many more applications. In fact, the methods and algorithms proposed here can be applied to all monitoring systems within the electric power systems hierarchy, from generation to transmission to all parts of the monitored distribution system.

References

[1] D. E. Whitehead, K. Owens, D. Gammel and J. Smith, "Ukraine cyber-induced power outage: Analysis and practical mitigation strategies," *2017 70th Annual Conference for Protective Relay Engineers (CPRE)*, College Station, TX, 2017, pp. 1-8. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8090056&isnumber=8089819

[2] J. Liang, L. Sankar, and O. Kosut, "Vulnerability analysis and consequences of false data injection attack on power system state estimation" IEEE Transactions on Power Systems, vol. 31, no. 5, pp. 3864-3872, 2016.

[3] Y. Yuan, Z. Li and K. Ren, "Quantitative analysis of load redistribution attacks in power systems," Parallel and Distributed Systems, IEEE Transactions on, vol. 23, no. 9, pp. 1731–1738, Sept 2012.

[4] L. Jia, J. Kim, R. J. Thomas, and L. Tong, "Impact of data quality on real-time locational marginal price," IEEE Trans. Power Systems, vol. 29, no. 2, pp. 627–636, 2014.

[5] R. Podmore, "Digital Analysis of Power System Networks," 1972. [Online]. Available: https://ir.canterbury.ac.nz/bitstream/handle/10092/6164/podmore_thesis.pdf;jsessionid=1AE212C95058918B8CE937835F6CEB36?sequence=1

[6] Department of Energy, "Contribution to Power system state estimation and transient stability analysis," 1984. [Online]. Available: https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/DE84014170.xhtml

[7] Lightbend 2018, "Play Framework,", https://www.playframework.com/

[8] Mike Bostock, "D3.js library,", https://d3js.org/

[9] The Cytoscape Consoritum, "Cytoscape.js library", http://js.cytoscape.org/

[10] J. Salmeron, K. Wood, and R. Baldick, "Analysis of electric grid security under terrorist threat," IEEE Transactions on Power Systems, vol. 19, no. 2, pp. 905–912, May 2004.

[11] D. Alderson, G. Brown, W. Carlyle, and R. Wood, "Solving defender-attacker-defender models for infrastructure defense," in 12th INFORMS Computing Society Conference, 2011.

[12] K. Arrow and G. Debreu, "Existence of equilibrium for a competitive economy," Econometrica, vol. 22, pp. 265–290, 1954.

[13] L. Mathiesen, "Computation of economic equilibria by a sequence of linear complementarity problems," Mathematical Programming Study, vol. 23, pp. 144–162, 1985.

[14] R. Andreani and J. M. Martinez, "On the solution of mathematical programming problems with equilibrium constraints," Mathematical Methods of Operations Research, vol. 54, no. 3, pp. 345–358, 2001.

[15] M. C. Ferris, S. P. Dirkse, and A. Meeraus, "Mathematical programs with equilibrium constraints: Automatic reformulation and solution via constrained optimization," 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.7017

[16] D. W. H. Cai, S. Bose, and A. Wierman, "On the role of a market maker in networked cournot competition," CoRR, 2017. [Online]. Available: http://arxiv.org/abs/1701.08896

[17] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," Operations Research, vol. 8, pp. 101–111, 1960.

[18] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," Numerische Mathematik, no. 4(3), pp. 238–252, September 1962.

[19] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids” in Proceedings ofthe 16th ACM Conference on Computer and Communications Security, ser. CCS 09. New York, NY, USA: ACM, 2009, pp. 2132.

[20] G. Hug and J. A. Giampapa, “Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks” IEEE Transactions on Smart Grid, vol. 3, no. 3, pp. 1362-1370, Sept 2012.

[21] M. E. Lubbecke, “Column generation,” Wiley Encyclopedia of Operations Research and Management Science (EORMS), 2010.

[22] I. Muter, ¸ S. ˙I. Birbil, and K. Bülbül, “Simultaneous column-and row generation for large scale linear programs with column-dependent rows,” Mathematical Programming, vol. 142, no. 1, pp. 47–82, 2013.

[23] “Electric Reliability Council of Texas (ERCOT) nodal protocols,” January 2015. [Online]. Available: http://ercot.com/mktrules/nprotocols/2015/02/February_2,_2015_Nodal_Protocols.pdf

[24] J. F. Benders, “Partitioning procedures for solving mixed-variables programming problems,” Numerische Mathematik, no. 4(3), pp. 238–252, September 1962.

[25] A. J. Conejo, R. Minguez, E. Castillo, and R. Garcia-Bertrand, Decomposition Techniques in Mathematical Programming. Springer

[26] M. A. Rahman and H. Mohsenian-Rad, “False data injection attacks with incomplete information against smart power grids,” in 2012 IEEE Global Communications Conference (GLOBECOM), Dec 2012, pp. 3153–3158.

[27] X. Liu and Z. Li, “Local load redistribution attacks in power systems with incomplete network information,” IEEE Transactions on SmartGrid, vol. 5, no. 4, pp. 1665–1676, July 2014.

[28] ——, “False data attacks against AC state estimation with incomplete network information,” IEEE Transactions on Smart Grid, vol. PP, no. 99, pp. 1–10, 2016.

[29] “IEEE 118-bus, 54-unit, 24-hour system, unit and network data.” [Online]. Available: motor.ece.iit.edu/data/JEAS_IEEE118.doc

[30] J. Kim and L. Tong, “On topology attack of a smart grid: Undetectable attacks and countermeasures,” IEEE J. Sel. Areas Commun., vol. 31, no. 7, pp. 1294–1305, Jul. 2013.

[31] J. Zhang and L. Sankar, “Implementation of unobservable state preserving topology attacks,” in Proc. North Amer. Power Symp. (NAPS), Charlotte, NC, USA, Oct. 2015, pp. 1–6.

[32] PJM. (11 Nov 2017). PJM Metered Load Data [Online]. Available: https://www.pjm.com/markets-and-operations/ops-analysis/historicalload-data.aspx

[33] L. Liu, M. Esmalifalak, Q. Ding, V. Esmesih, and Z. Han, “Detecting False Data Injection Attacks on Power Grid by Sparse Optimization”, IEEE Transactions on Smart Grid, vol. 5, no. 2, pp. 612-621, March 2014.

[34] J. Zhao, G. Zhang, and R. Jabr, “Robust Detection of Cyber Attacks on State Estimators Using Phasor Measurements”, IEEE Transactions on Power Systems, vol. 32, no. 3, pp. 2468-2470, May 2017.

[35] M. Ozay, I. Esnaola, F. Vural, S.Kulkarni, H. V. Poor, “Machine Learning Methods for Attack Detection in the Smart Grid”, IEEE Transactions on Neural Networks and Learning Systems, Vol. 27, No. 8, August 2016.

[36] T. Cover, P. Hart, “Nearest neighbor pattern classification”, IEEE Transactions on Information Theory, Vol. 13, Issue 1, January 1967.

[37] Varun Chandola, Arindam Banerjee, and Vipin Kumar, “Anomaly Detection: A Survey”, ACM Computing Surveys, Vol. 41(3), Article 15, July 2009.
[38] G. Prajapati, and A. Patle, “On Performing Classification Using SVM with Radial Basis and Polynomial Kernel Functions”, Emerging Trends in Engineering and Technology (ICETET), 2010 3rd International Conference, Nov. 2010.
[39] Levenberg, Kenneth (1944). “A Method for the Solution of Certain NonLinear Problems in Least Squares”. Quarterly of Applied Mathematics. 2: 164-168.
[40] Marquardt, Donald (1963). “An Algorithm for Least-Squares Estimation of Nonlinear Parameters”. SIAM Journal on Applied Mathematics. 11 (2): 431-441.

Part II

Anomaly detection for wide-area protection and control in smart grid

Manimaran Govindarasu

Graduate Students
Vivek Kumar Singh
Pengyuan Wang

Iowa State University

For information about this project, contact

Manimaran Govindarasu
Professor, Department of Electrical and Computer Engineering
Iowa State University
Ames, Iowa, USA 50011-3060
Phone: 515-294-9175
Fax: 515-294-3637
Email: gmani@iastate.edu

Power Systems Engineering Research Center

The Power Systems Engineering Research Center (PSERC) is a multi-university Center conducting research on challenges facing the electric power industry and educating the next generation of power engineers. More information about PSERC can be found at the Center's website: http://www.pserc.org.

For additional information, contact:

Power Systems Engineering Research Center
Arizona State University
527 Engineering Research Center
Tempe, Arizona 85287-5706
Phone: 480-965-1643
Fax: 480-727-2052

Notice Concerning Copyright Material

Table of Contents

1. Introduction 1
   1.1 Background 1
   1.2 Overview of the Problem 2
   1.3 Report Organization 2
2. Wide Area Protection and Control (WAPAC) 4
   2.1 Introduction to WAPAC 4
      2.1.1 Remedial Action Scheme (RAS) 4
      2.1.2 Automatic Generation Control (AGC) 4
   2.2 Cyber Attack Taxonomy 5
3. Cyber Attack Vectors 7
   3.1 Single Attack Classification 7
      3.1.1 Malicious Tripping 7
      3.1.2 Pulse Attack on Generator 7
      3.1.3 Ramp Attack on Generator 7
      3.1.4 Malware based Attack 8
      3.1.5 Replay Attack 8
   3.2 Coordinated Attack Vectors 8
4. Anomaly Detection for Remedial Action Scheme 9
   4.1 Generation Rejection RAS 9
   4.2 Machine learning based Anomaly Detection 10
      4.2.1 Decision Tree (Machine Learning) based Proposed Methodology 10
      4.2.2 Experimental Setup and Case Study 11
      4.2.3 Results and Discussions 13
      4.2.3.1 Offline and Real-time Testing 13
   4.3 Model based Anomaly Detection 14
      4.3.1 Proposed Approach and Implementation 14
      4.3.1.1 Cyber Attack vector 14
      4.3.2 Proposed Approach for IDS 15
      4.3.3 IDS Implementation 17
      4.3.4 Experimental Setup 17
      4.3.5 Results and Discussions 18

4.3.5.1 Performance Evaluation of IDS 18
4.4 Multi-Agent based Anomaly Detection 18
4.4.1 Coordinated Attack Vectors 19
4.4.2 Proposed Architecture and Methodology 19
4.4.2.1 Proposed Multi-Agents based Hierarchical Architecture 19
4.4.2.2 Proposed Anomaly Detection Algorithm 21
4.4.3 Experiment Setup and Case Studies 22
4.4.4 Results and Discussions 24
4.4.4.1 Performance Evaluation 24
5. Anomaly Detection for Automatic Generation Control 25
5.1 Basics of AGC 25
5.1.1 Conventional AGC 25
5.1.2 Proportional Integral Derivative (PID) ACE based AGC 25
5.2 Model based Anomaly Detection 25
5.2.1 Cyber Attack Vector 26
5.2.1.1 Ramp Attack 26
5.2.2 Attack Detection and Mitigation 26
5.2.3 Experimental Implementation 27
5.2.4 Performance Evaluation 28
5.2.4.1 Attack Analysis 28
5.2.4.2 Defense Analysis 31
5.2.4.3 Effect of Mitigation on AGC Performance 31
5.3 Machine Learning based Anomaly Detection 32
5.3.1 Abnormal Generation Control Detection 33
5.3.1.2 Overall Anomaly Detection Process 33
5.3.2 Semi-Supervised Clustering with HDBSCAN 33
5.3.3 Experimental Results 34
5.3.3.1 Introduction of the Datasets 34
5.3.3.2 Clustering with K-means 35
5.3.3.3 Clustering with Div_Conq_HDBSCAN 36
6. Conclusions 38
References 39

List of Figures

Figure 1. Attack surfaces in generic WAMPAC architecture. 5
Figure 2. Single and coordinated attacks in time and space 8
Figure 3. Generation rejection RAS architecture 9
Figure 4. Proposed methodology for ADS in Remedial Action Scheme (RAS). 10
Figure 5. Experimental setup for the proposed methodology. 12
Figure 6. Generated decision trees (DT) for 60% training data (case 4) 13
Figure 7. Real-time testing of intelligent RAS 14
Figure 8. Steps involved in creating coordinated attacks on RAS. 15
Figure 9. Generic architecture of behavior-rule based ADS (BRADS). 15
Figure 10. Proposed intrusion detection engine for the ramp attack. 16
Figure 11. Alert examples in the log files of Snort and Bro IDS. 17
Figure 12. Experimental setup for attack implementation and detection 17
Figure 13. Detection rate (a) and average latency (b) of alert messages for Bro and Snort IDS. 18
Figure 14. Overview of the proposed architecture for multi-agent RAS. 20
Figure 15. Anomaly detection methodology to detect the compromised RASc. 21
Figure 16. Experimantal set-up for attack implementation in PowerCyber Lab. 22
Figure 17. Decentralized RAS enabled modified IEEE 39 bus system. 23
Figure 18. Online anomaly detection topology for the 3 zones RAS. 23
Figure 19. Detection Cycles (a) and Latecy cycles (b) for pulse and ramp attack. 24
Figure 20. Algorithm for attack detection and mitigation. 26
Figure 21. Procedure to determine ACE bounds for mitigation. 27
Figure 22. IEEE 39 bus model. 28
Figure 23. Attack comparison for different levels of renewables. 28
Figure 24. Attack comparison between Conventional & PID based AGC. 29
Figure 25. Performance comparison between Conventional & PID based AGC. 30
Figure 26. Effect of mitigation algorithm on attack. 31
Figure 27. AGC performance with mitigation algorithm. 32
Figure 28. Generation control just before AGC operation. 33
Figure 29. Proposed algorithm for semi-supervised clustering. 34
Figure 30. IEEE 39 bus model divided into 3 Bas. 35
Figure 31. Training datasets. 35

Figure 32. Clustering results with K-means. 36
Figure 33. Clustering results with HDBSCAN. 37
Figure 34. Test confusion matrix of K-Means Clustering (TABLE II) and proposed clustering (TABLE III). 37

List of Tables

Table 1. Differential PMU features 11
Table 2. Training and testing for different datasets 13
Table 3. Roles assigned to different agents 20
Table 4. Attack impact analysis 30
Table 5. Training and testing datasets for AGC 34

1. Introduction

1.1 Background

The modern power grid is a complex cyber-physical system that is being increasingly integrated with smart sensors, advanced communication infrastructures, and sophisticated analytics and controls for achieving improved efficiency, reliability, and resiliency for the grid. Although the cyber-based technologies are among the key drivers of grid modernization, they also increase the number of digital access points the grid has and hence its attack exposure [1]. Recent literature and government documents have highlighted the need of secure networks for the SCADA based critical infrastructure like the power grid, which is increasingly becoming a constant target of cyber related attacks [2]. In recent years, several malicious cybersecurity incidents have been reported, by the Industrial Control Systems Cyber Emergency Response Team (ICS-CERT), target the industrial control systems [3]. The paper in [4] provides the detailed documentation related to the several cybersecurity incidents related to the power system infrastructures. Based on the analysis, it highlights the observations that the attacks are happening frequently, many of the attacks have disrupted normal operation and the attackers are operating in the stealth mode through malware attacks, social engineering, etc. Stuxnet, the incredible complex computer worm, which has targeted the industrial control system by modifying the programmable logic controllers, has affected more than 100,000 industrial devices by the year 2010 [5]. In October 2012, a power company reported a virus infection in the turbine control system which delayed the plant restart for multiple weeks [5]. The recent hack of Ukraine's power grid is considered a sophisticated malware based coordinated attacks in the SCADA environment which caused shutdown of 7 110 kV and 23 35 kV substations for three hours [6]. The incident is the first known and officially reported cyber-attack causing the power outages. Furthermore, several remote code vulnerabilities have been issued in the past, e.g., CVE-2017-7494, CVE-2016-7855, CVE-2014-4114 vulnerabilities in windows operating system can allow backdoor access to the attacker [7]. In July 2018, the Department of Homeland Security (DHS) has issued the warning alerts against the international threat actors who had infiltrated the industrial control system in the past that could have caused grid blackouts [8].

The SCADA/EMS system provides essential functions as necessary through Wide-Area Protection and Control (WAPAC) applications for maintaining the stability and reliability of the power system. The WAPAC applications rely on state-of-the-art synchrophasor measurement technology and the existing Supervisory Control and Data Acquisition (SCADA) infrastructure to perform appropriate control action in real-time. NERC has classified the WAPAC as a critical asset with the cyber physical properties and any compromise and degradation in the scheme can affect the reliability and stability of the bulk power system [9]. Given the amount of conventional security measures deployed in the WAPAC which are also coupled with legacy infrastructure, it is not the matter of 'if" but a matter of 'when" regarding these existing wide area applications becoming exploited to the cyber-attacks. Therefore, there is a strong need to go beyond the traditional security solutions and develop more robust, efficient anomaly detection system (ADS) in the face of advanced, persistent adversaries.

There exist the limited research works which address the vulnerabilities in the communication networks as well as possible cyber-attacks on WAPAC applications [10], [11]. Hahn et al.

performs the vulnerability assessment on SCADA communication protocols and, shows how the coordinated denial of service (DoS) and malicious tripping attack on the wide-area protection scheme can cause a generator outage while affecting the system's transient voltage stability [10]. Aditya et al. presents the experimental evaluation of the data integrity attacks on wide area generation control using cyber-physical testbed [12]. Anurag et al. shows how the attacker can leverage publicly available information and apply graph theory-based approach to find the vulnerabilities in the cyber-physical system [13].

Although very useful, none of these works completely addresses the vulnerabilities due to insider and outsider threats. The outsider threat like malicious tripping attack can open circuit breakers at an inappropriate time, which can trigger the WAPAC controller to take unnecessary actions. Similarly, insider threat like phishing attack can cause sophisticated malware installation in the system, which can penetrate the communication network internally providing unauthorized access to the attacker from outside the network's perimeter. Once the attacker gains access to the network, they can perform different classes of attacks like reconnaissance, denial of service, data integrity, etc. Although, it seems difficult to inject malware inside the control center without getting detected due to high security network, it can be propagated within geographically distributed devices installed in substations. Therefore, the in-depth analysis of security threats is required for the development of efficient Anomaly Detection System (ADS) against stealthy, sophisticated attacks. Based on the redundant measurements, known trails of system abnormal behavior during attacks and applying advanced machine learning techniques, it is possible to identify the anomalies in the context of WAPAC cyber physical security.

1.2 Overview of the Problem

In this project, our primary goal is to develop the efficient and robust Anomaly Detection System (ADS) to detect possible cyber-attacks which can have a significant impact on the grid stability in the context of WAPAC applications. Since, there exists no single technology that can optimally provide promising solutions to the existing multiple level security problems at the cyber-physical layer, we have leveraged different existing techniques including machine learning, behavior models and redundant measurements (SCADA, PMUs, historical data) to develop the next-generation ADS which can overcome the limitations of conventional security solutions. Specifically, we have focused on decision trees, semi-supervised clustering-based machine learning, temporal behavior based models and multi-agent based architectures and methodologies in developing novel anomaly detection engines. Furthermore, we have also performed the impact analysis for possible cyber-attacks, including single and coordinated attack vectors, to comprehend their impacts on the system performances. As a proof of concept, we have implemented the proposed solutions through the experimental setup using the resources of PowerCyber CPS security testbed available at Iowa State University and evaluated their performances through the real-time testing.

1.3 Report Organization

The report is organized as follows:

1. Section 2 talks about the Wide Area Protection and Control (WAPAC) and its two main components: Wide Area Protection (WAP), commonly known as Remedial Action Scheme

(RAS), and Wide Area Control, especially Automatic Generation Control (AGC). It also illustrates the possible attack surfaces based on the existing components of the generic architecture.

2. Section 3 describes the various types of attacks which we have considered in WAPAC application. It discusses the attacks in detail and provides the mathematical expressions to understand the nature of attacks, their possible targets and how these attack vectors can be modelled or implemented in the control applications.

3. Section 4 describes the different types of anomaly detection in the context of WAP scheme. Initially, it discusses about the generation rejection RAS, which sheds the generation to prevent overloading on the connected transmission lines. Based on the nature of attacks, either single or coordinated, machine learning, temporal behavior based model, and multi-agent based architectures are proposed to detect different flavors of attacks. The proposed solutions are also implemented in cyber physical environment and tested in real-time platform.

4. Section 5 presents the different types of the anomaly detection for Automatic Generation Control (AGC). We have considered two types of anomaly detection: model based ADS and machine learning based AGC. The model-based ADS leverages the historical data as the redundant measurement to create different types of bounds/rules to detect attacks. The machine learning based ADS deploys semi-supervised clustering algorithm for detecting cyberattacks. The performance of the proposed approach is also compared with other existing algorithms.

2. Wide Area Protection and Control (WAPAC)

2.1 Introduction to WAPAC

The WAPAC is the essential component of Energy Management System which leverages the state-of-the-art synchrophasor measurement technology and existing SCADA infrastructure to provide the real-time protection and control of the power system. The WAPAC relies on the data sharing devices and communication network to provide the timely control operations, hence the security of appropriate cyber infrastructure is vital. WAPAC can be divided into its two constituent components: Wide Area Protection (WAP), and Wide Area Control (WAC). The WAP is also known as Remedial Action Scheme (RAS), or Special Protection Scheme (SPS), collects the system information over the wide geographic area and provides appropriate corrective actions to mitigate the large disturbances, and maintains the stability of the grid. Wide Area Control like Automatic Generation Control (AGC) relies on the SCADA infrastructure to balance the generation-load demand and limit the tie-line power flows between multiple Balanced Areas (BAs).

2.1.1 Remedial Action Scheme (RAS)

Remedial Action Scheme (RAS) is a protection scheme needed to secure the system during disturbances. Since the power system is not robust enough to accept components failures without subsequent response, small disturbances in the system may affect the voltage stability and may lead to cascading failure. It detects the physical disturbances like line outages, generator outages and later performs corrective actions like generation shedding, load shedding and other defined actions to maintain the system's stability and reliability. According to the NERC definition of RAS scheme, corrective actions during abnormal conditions include changes in demand, generation (MW or MVAR) as well as system configuration to maintain power flows and system stability [14]. It is also allowed to perform system restoration (auto-reclosing) along with corrective actions to minimize impact on system and restoration efforts by system operators [14].

As the number of PMUs installed in substation keep increasing, their applications related to wide area monitoring and protection are gaining more popularity. The synchrophasor based Remedial Action Schemes (RASs) are also implemented in the real world to perform autonomous corrective actions and maintain the system stability during the component failures. The PERC report in [15] talks about PMU based RAS schemes deployed in utilities and companies such as BPA, SCE, etc. The paper in [16] talks about the PMU based RAS for predicting the catastrophic events in the power system.

2.1.2 Automatic Generation Control (AGC)

The modern power system is divided into multiple Balancing Authorities or Balancing Areas (BAs) which are connected through bulk transmission lines and each BA is responsible for maintaining its regional generation and load demand to the maintain the system nominal frequency (60 Hz). Automatic Generation Control (AGC) operates at the control center and is the critical component of Energy Management System. It is the wide-area secondary controller, which is used for maintaining the system nominal frequency by balancing the generation with the load as well

as maintaining the tie line flows between the balancing areas [17]. The scheme employs Area Control Error (ACE) calculated from the frequency and tie line flow deviations every 2-8 seconds as shown in equation 2.1. In this equation, ΔP_{tie} represents the difference in the tie line flows from the actual tie-line flows, $P_{act,}$ and schedule tie-line power flows, P_{sch}. Apart from the tie-line flow deviations, ACE also relies on Δf, which is the difference between the actual frequency, f_{act} and nominal frequency, f_{nom}, and the balancing authority bias β. In the final equation 2.3, ΔP_{load} is the load change in the BA, D is the frequency sensitivity of loads, and R represents the operating generator in the given power system. Finally, the control signal generated from the computed ACE value at the control center is sent to the actuator to control the generation.

$$ACE = \Delta P_{tie} + \beta \Delta f \tag{2.1}$$

$$\Delta P_{tie} = P_{act} - P_{sch} \tag{2.2}$$

$$\Delta f = f_{act} - f_{nom} = -\frac{\Delta P_{load}}{\sum (1/R + D)} \tag{2.3}$$

2.2 Cyber Attack Taxonomy

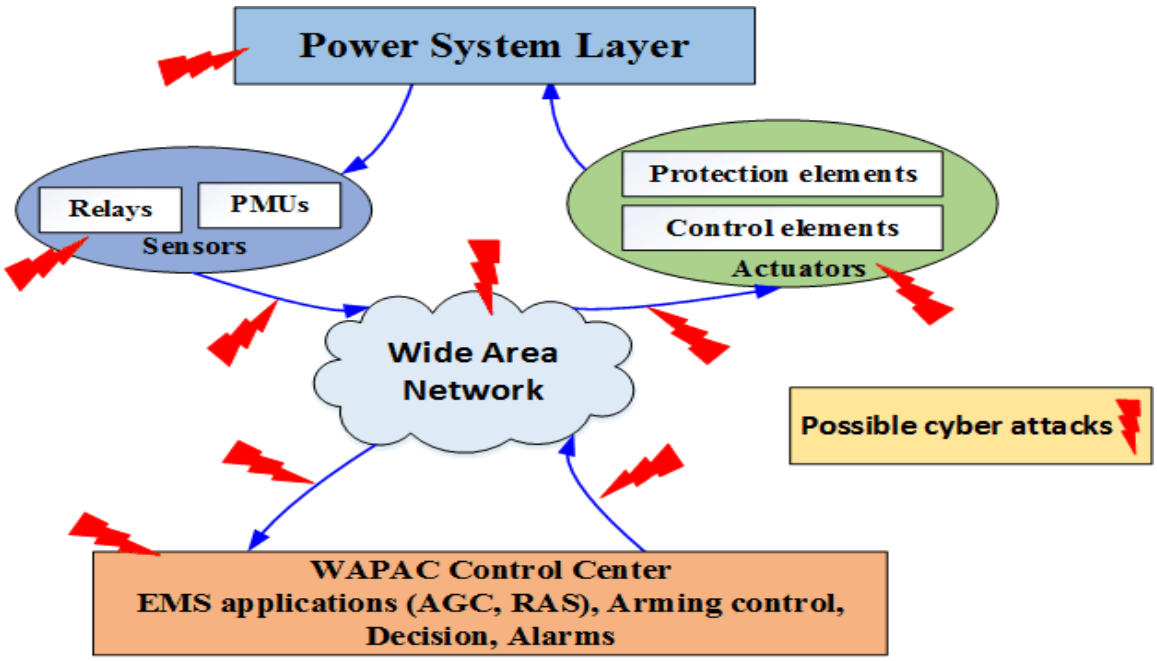


Figure 1. Attack surfaces in generic WAMPAC architecture.

Figure 1 shows the generic architecture of WAPAC, which consists of multiple components, including sensors, actuators and the high-level controller, which operates in the timely manner to maintain the power system stability. The power system data (currents, voltages, digital logs) are measured through the sensors and forwarded through the high-speed, wide-area communication

network to the WAPAC application in the control center. The controller takes appropriate corrective actions based on the system conditions by sending control signals to the actuator. The actuators consist of local protection elements and VAR control elements like FACTS and SVC for voltage related applications [11].

The integrated cyber infrastructure with data sharing devices shows possible attack surfaces which can be exploited by attackers. Since the existing WAPAC applications are not designed conventionally to handle failures related to cyber-attacks, the unexpected cyber related disturbances can affect the normal operation of the WAPAC as well as the grid stability. For example, the attackers can compromise the sensors and actuators to perform the data integrity attacks which may cause unnecessary generation/load shedding. The attacker can also sniff the network packets or perform attack reconnaissance which can be later exploited for performing stealthy Man-in-the Middle or denial-of-service (DoS) attacks. It may cause loss of observability and controllability for the operator sitting in the control center. The attacker can also compromise the controller, operating at the control center, through the stealthy malware to disrupt the normal operation and perform severe attacks like generation altering attacks or unexpected load shedding. Also, the advanced, persistent attackers can leverage their extensive resources and expert skills to perform multiple attacks in a coordinated fashion, making it difficult to be detected by the conventional security methods.

3. Cyber Attack Vectors

In general, the cyber-attacks can be classified into single and coordinated attacks. The single attack vectors include isolated attacks on the measurements, controls, wide-area communication and systems. It is performed by the attacker with limited skills, less capability and resources constrained. However, the coordinated attacks involve the combination of multiple attacks coordinated in time, and or space. It is performed by the advanced persistent attackers with the motive of performing severe impact on the system without getting detected. Since our main motivation is to develop the sophisticated anomaly detection, which can provide consistent performance irrespective of the nature of attacks, we have considered different types of cyber-attacks either single or coordinated, irrespective of attacker's intelligence, which can have an impact on the physical stability of the grid.

3.1 Single Attack Classification

3.1.1 Malicious Tripping

The malicious tripping attack can be performed in multiple different ways. Attackers can perform the tripping attack by getting unauthorized access to the control center. At substation level, setting of physical relays can be altered to cause tripping of breakers. Attackers can also perform the attack through the SCADA communication network. In this work, we have implemented the tripping attack by eavesdropping the network packets going between the substation and control center. Once the attacker has internal access to wide area network, he/she can easily learn about the network packets used to trip the relays and eventually replay the tripping packet to perform the attack.

3.1.2 Pulse Attack on Generator

This generation altering attack vector involves periodically changing the input control signal sent to the generator. In this attack, a control signal is modified by adding the pulse attack parameter, λ_{pulse}, for a small interval (t_1) and retaining back the original input for the remaining interval (T- t_1) for the given time period (T).

$$P_{pulse} = \begin{bmatrix} P_i(1+\lambda_{pulse}) & (t = t_1) \\ P_i & (t = T - t_1) \end{bmatrix} \tag{3.1}$$

3.1.3 Ramp Attack on Generator

This generation altering attack vector involves adding a time varying ramp signal to the input control signal sent to the generator. The ramp signal is decided based on the ramp signal parameter, λ_{ramp}.

$$P_{ramp} = P_i + \lambda_{ramp} * t \tag{3.2}$$

3.1.4 Malware based Attack

The term malware is defined for any software having malicious, malevolent intention. It can be installed in the controller through any openings (like email, USB, social engineering, etc.). The malware creates a backdoor once it is installed, which provides an access to the system from any outside network. After getting the unauthorized access, the attacker can read, modify or delete the logic program running for the controller operation on the system.

3.1.5 Replay Attack

The replay attack involves eavesdropping the network/data packets on the grid network and later injecting deceptive measurements disguised as genuine measurements to misguide the operator. It can be expressed mathematically, where, X_r represents the instant measurement sent to operator at time t. During the replay attack, X_r is modified to the deceptive/old measurement, x_i, and, x_n is the measurement during the normal operating condition without any attack.

$$X_r = \begin{bmatrix} x_i \ (t = t_1) \\ x_n \ (t \neq t_1) \end{bmatrix} \tag{3.3}$$

3.2 Coordinated Attack Vectors

The coordinating attacks are the combination of multiple attacks which are coordinating in time, and/or space to perform severe impact on the physical system as well as disguising their malicious actions from getting detected by the operator or conventional security solutions. Figure 2 shows how the single attacks can be leveraged to perform highly sophisticated attacks. The recent hack of Ukraine's power grid is the real coordinated attack in the SCADA environment which caused shutdown of 7 110 kV and 23 35 kV substations for three hours.

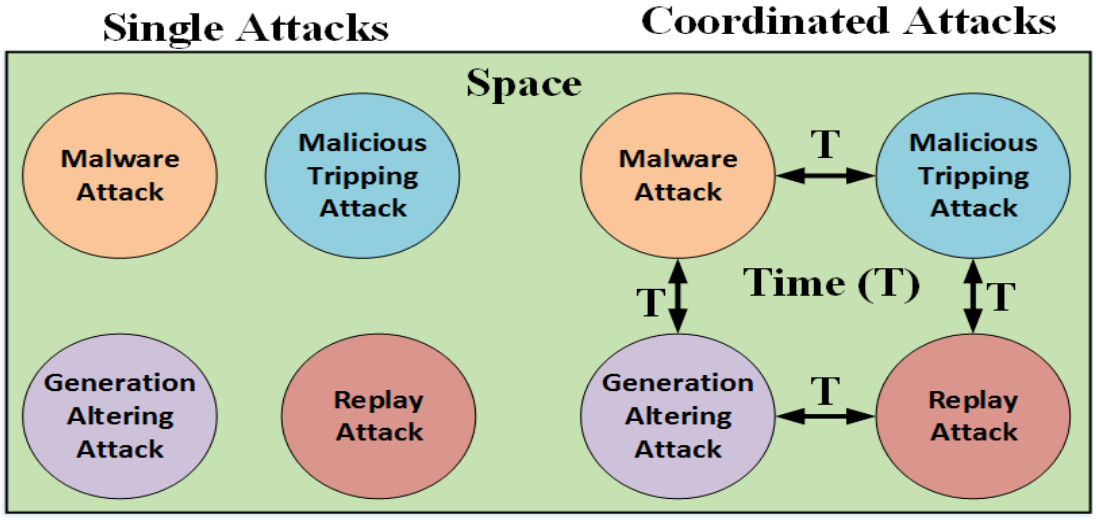


Figure 2. Single and coordinated attacks in time and space

4. Anomaly Detection for Remedial Action Scheme

4.1 Generation Rejection RAS

There are different types of remedial action schemes deployed in the power industries. In this project, we have focused on the generation-rejection based remedial action scheme which sheds the generation during the line contingency to prevent the thermal overloading on the transmission lines. Figure 3 shows the generic architecture of generation rejection RAS. It polls the data using the sensors (relays, PMUs) at regular intervals in terms of relays status, the line flows and power output of the generator. Once the line contingency is detected through relays, the RAS controller is activated. It checks the operational transfer capacity (OTC) limit of the adjacent transmission lines directly connected to the generator. If the power flows in the connected, adjacent transmission lines exceed the defined OTC maximum limit (OTC_max limit), the RAS controller will be enabled. It will shed the required amount of generation as defined by the action table. We have considered the thermal limit of transmission lines while computing the OTC limit. We have also included the auto-reclosing scheme based on the NERC definition of RAS. Figure 3 also shows the possible attack surfaces which can be exploited by the attackers. More details are provided in the paper [18].

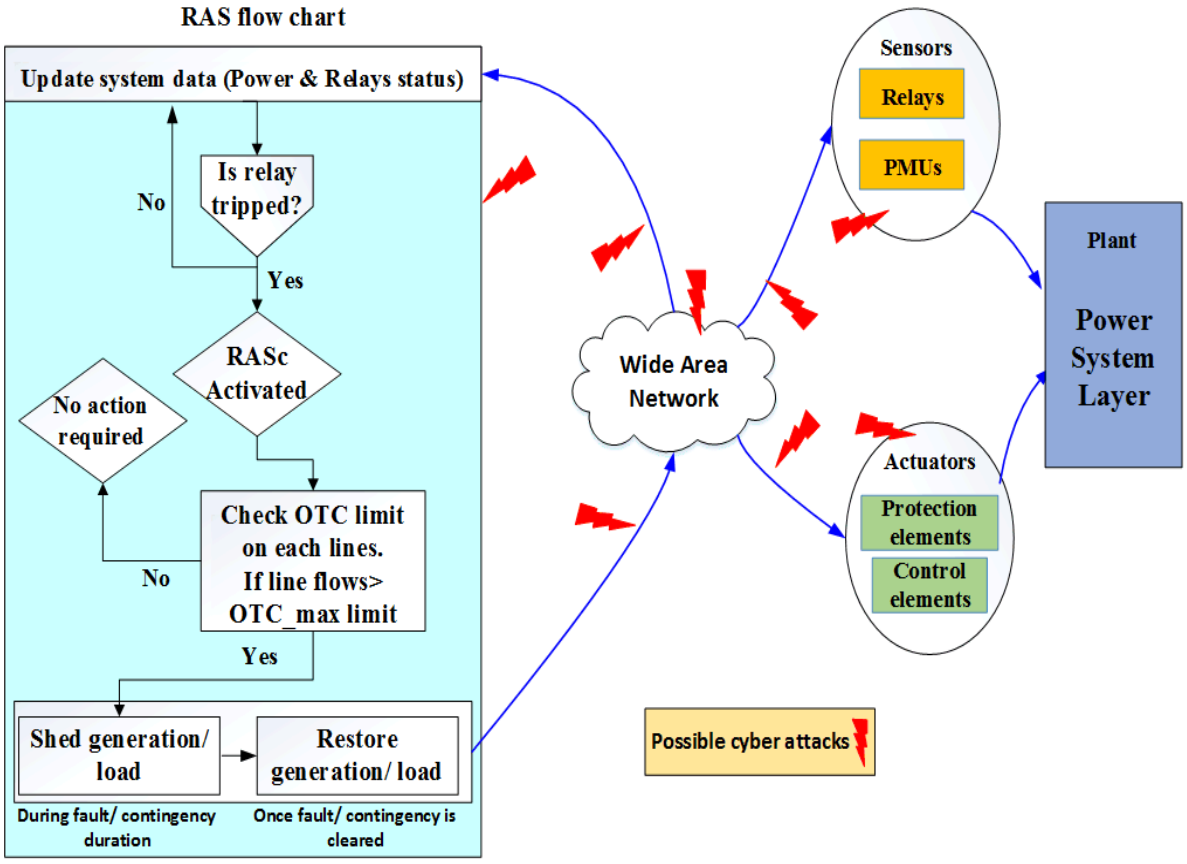


Figure 3. Generation rejection RAS architecture

4.2 Machine learning based Anomaly Detection

In this work, we have proposed the decision tree-based anomaly detection for detecting malicious tripping attack in the real-time. We have leveraged the phasor measurement units (PMUs) to build the decision tree rules with an assumption that the PMU measurements are secure. We have computed the differential features of voltage and current phasors of the sending and receiving end of transmission lines which are later deployed in building the classification model. Finally, we have implemented the proposed methodology using the cyber physical testbed on modified IEEE 39 bus system and performed offline and real-time testing to evaluate its performance. More details of this work are provided in the project publication [9].

4.2.1 Decision Tree (Machine Learning) based Proposed Methodology

The Figure 4 shows the proposed methodology for ADS to develop the intelligent RAS, which can overcome the limitation of the conventional RAS by predicting the malicious and legitimate behavior of relays. In this case, we are considering the malicious relay tripping attack as the main attack vector and developing the ADS to detect and distinguish it from the normal tripping during the line fault. Initially, it collects the data from PMUs periodically through the analog and phasor measurements, and during the tripping of relays, it checks with the anomaly detection engine (ADE) to predict the malicious and normal tripping. In case of normal tripping, it sheds the

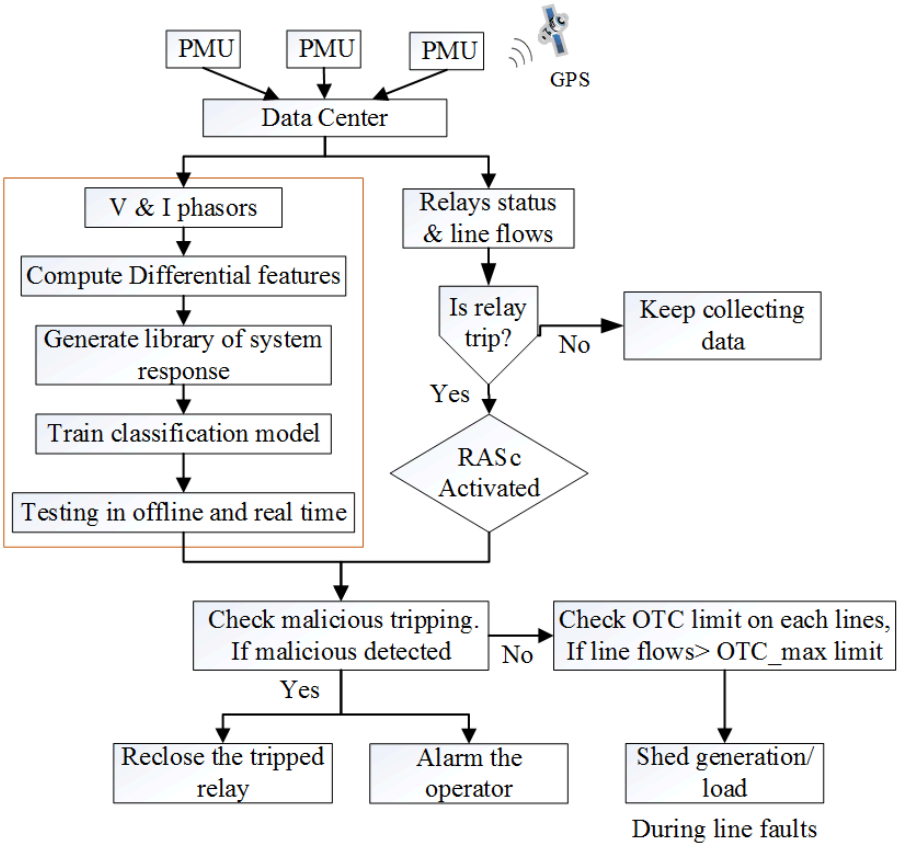


Figure 4. Proposed methodology for ADS in Remedial Action Scheme (RAS).

Table 1. Differential PMU features

Vars	*Features*	*Description*
X1	VMs − VMr	Positive sequence voltage magnitude (VM) difference
X2	∂(VMs − VMr)/ ∂t	Rate of change of positive sequence VM difference
X3	VAs − VAr	Positive sequence voltage angle (VA) difference
X4	∂(VAs − VAr)/ ∂t	Rate of change of positive sequence VA difference
X5	IMs – IMr	Positive sequence current magnitude (IM) difference
X6	∂(IMs – IMr)/∂t	Rate of change of positive sequence IM difference
X7	(IAs – IAr)	Positive sequence current angle (IA) difference
X8	∂(IAs – IAr)/∂t	Rate of change of positive sequence IA difference

generation as usual based on the predefined action table, however, when the malicious tripping is detected, it recloses the tripped relays to avoid overloading on other lines instead of shedding the generation and load. It also sends the alarm alertness to the operator to provide situational awareness. The colored box of Figure 4 shows the different steps involved in developing the ADE in terms of input selection, building, training and testing of DT. Table 1 shows 8 differential features which we have extracted for building the classification model where subscripts s and r represent the sending and receiving end of the transmission lines.

4.2.2 Experimental Setup and Case Study

Figure 5 shows the experimental setup for implementing the attacks as well as testing the proposed methodology in offline and real-time testing mode. The modified IEEE 39 bus system is modeled in ePHASORsim and simulated in real-time digital simulator, OPAL-RT. The simulator is integrated with two physical relays which are connected to the remote terminal unit (RTU) inside the substation and are monitored, controlled by the control center. For attack implementation, we have captured the tripping packet going from the control center to the substation RTU using Wireshark and then replayed the captured packet to the remote terminal unit (RTU) using python script to trip the relays. We have used virtual PMUs modeled inside the OPAL-RT for generating phasors at 60 samples per second. The SEL 2407, the satellite synchronized clock, is providing time synchronization to the virtual PMUs in the simulator. The virtual PMUs are sending phasors and their differential features using IEEE C37.118 protocol to the iPDC, the phase data concentrator, in real-time. The iPDC is saving data to the MySQL database. Initially, the generated database is used for training and building the decision tree and further performing offline testing for different cases using rattle. During the real-time testing, the RASc is running in the python script which pulls the data coming from the simulator to the MySQL database in terms of line flows, relays status and differential PMU features. When the line is out, it checks with the decision

tree rules to identify events and performs control action by sending control signal directly to the simulator using DNP3 OPC server client protocol.

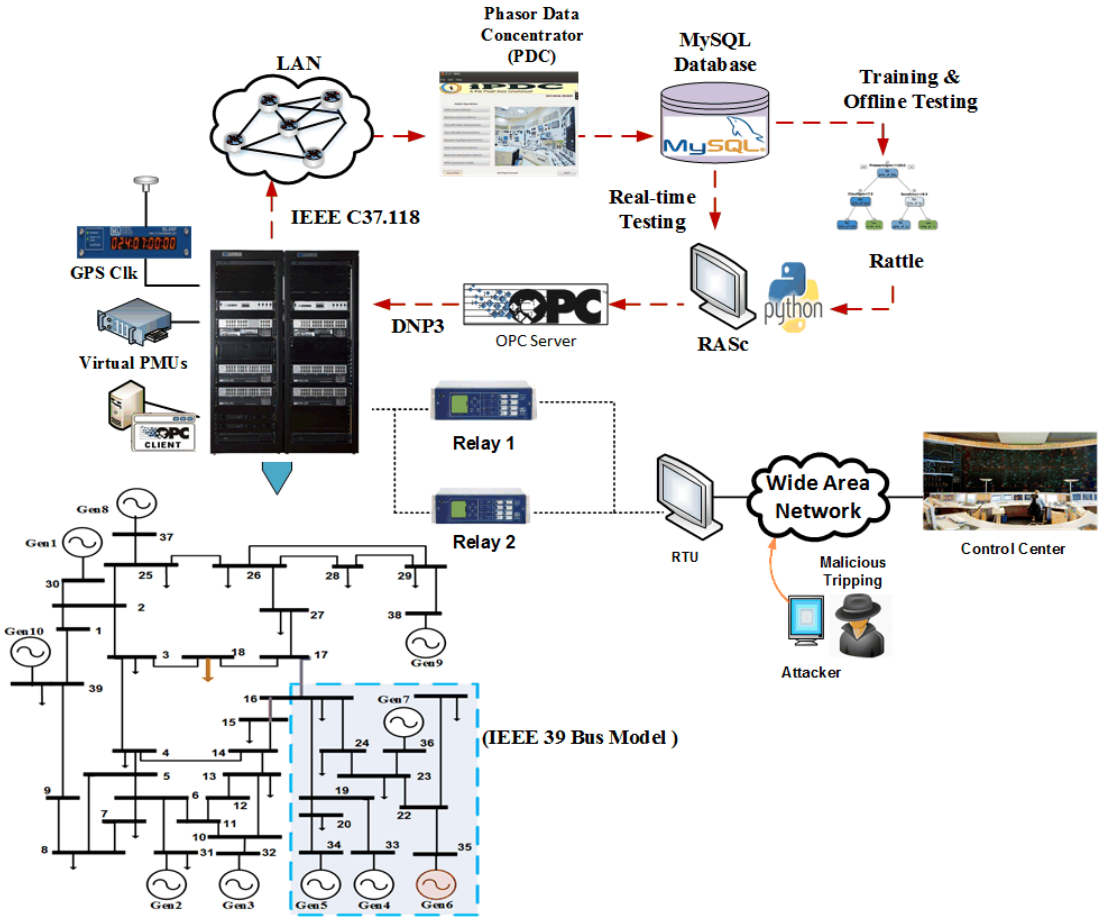


Figure 5. Experimental setup for the proposed methodology.

We have employed the modified IEEE 39 bus system which is divided into two major areas. The outlined Area1 is the primarily generation area which is supplying power to the rest of the system through the tie lines 15-16 (L15-16) and 16-17 (L16-17). To prevent thermal overloading on line L15-16, during the line outage L16-17, RASc sheds the generation at bus 35 and the equal amount of load is shed at bus 18 to maintain the load generation balance. We have computed the differential PMU features using bus 16 and 17 as sending and receiving end. We have created miscellaneous operating points through generation and load scaling. The generation at bus 35 is varied from 610 MW to 700 MW and load is varied from 118MW to 208 MW in equal step increase of 10 MW to maintain the generation and demand balance. For each operating point, we have simulated a 3 phase to ground fault followed by line tripping as normal tripping event and sudden line outage as a cyber attack event at line 16-17. We have varied the duration of the fault with mean values of 6 cycles and 0.667 cycles standard deviation. The fault location distance factor is varied from 0 to 1 along the length of line with step size of 0.1, excluding the limits (0 and 1). Total number of fault cases are 50 fault durations * 9 fault locations *10 operating points = 4500 cases. We have also simulated 10 line outages as the tripping attack, one for each operating point, and finally, total 4510 are simulated for the proposed method.

4.2.3 Results and Discussions

4.2.3.1 Offline and Real-time Testing

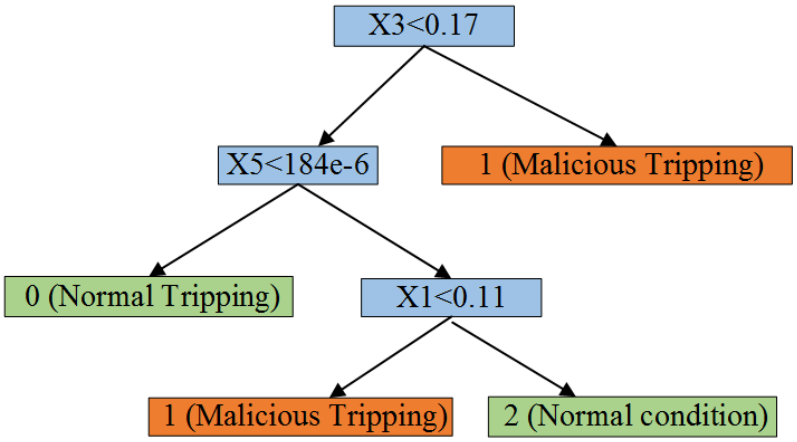


Figure 6. Generated decision trees (DT) for 60% training data (case 4)

Table 2. Training and testing for different datasets

Cases	***Training***	***Testing***	***Accuracy***	***Processing Time***
1	20	80	98.6	0.01
2	40	60	98.4	0.01
3	50	50	99.6	0.02
4	60	40	100	0.02
5	80	20	100	0.02

Table 2 shows the offline performance of the decision tree for different test cases in terms of accuracy and the processing time of training the model. It is obvious to note that the 60% training of data is enough to achieve 100% accuracy. Figure 6 shows the generated decision tree rules for 60% training data sets. It can be observed that only 2 features are sufficient to generate the detected rules. Figure 7 (a) and (b) shows the performance of the intelligent RAS during real-time testing in the cyber physical testbed. In this case, the intelligent RAS has detected the tripping attacks and reclose the relay in a short time span instead of shedding the generation.

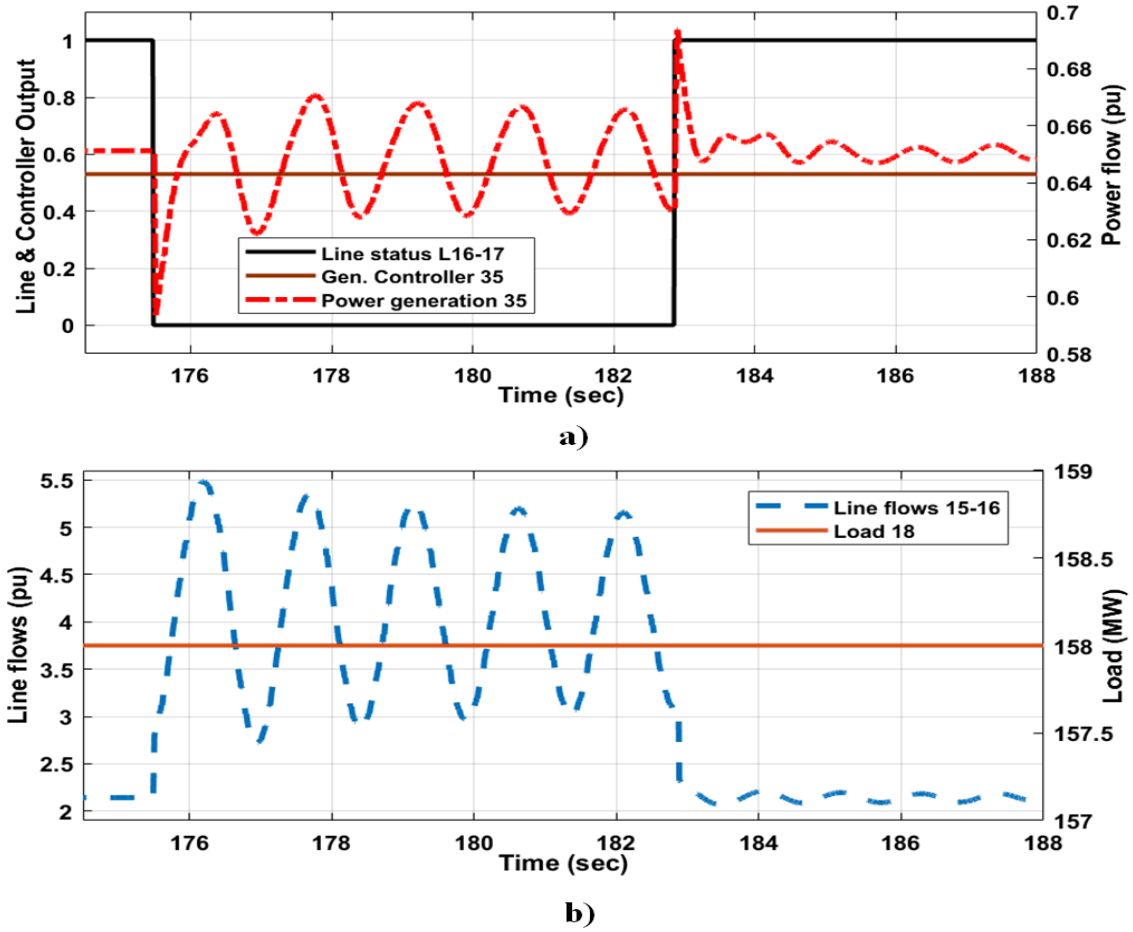


Figure 7. Real-time testing of intelligent RAS

4.3 Model based Anomaly Detection

In this work, we have proposed the temporal behavior-based ADS to detect the generation altering attacks in the context of RAS. We have implemented the proposed IDS using publicly available IDS tools Snort, BRO and compared their performance in terms of detection rate and alert latency in the cyber physical environment. More details of this work are provided in the project publication [7].

4.3.1 Proposed Approach and Implementation

4.3.1.1 Cyber Attack vector

We have implemented the coordinated attacks which involve installing malware on the RAS controller that closes the legitimate RAS program and run the malicious program. The malicious program initiates the generation altering attack through the ramp attack that involves slowing reducing the generation making it difficult to be detected by the conventional IDS. In order to disguise the ramp attack from getting detected, the attacker is also sending false measurement updates (replay attack) to the operator sitting in the control center, as shown in Figure 8.

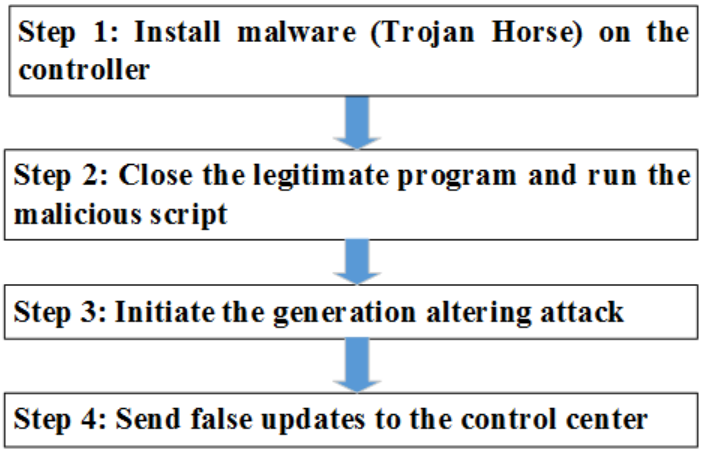


Figure 8. Steps involved in creating coordinated attacks on RAS.

Since the attacker has hacked the controller, it is reasonable to say that the implemented attack is difficult to be detected using access control and protocol whitelisting IDS. Therefore, we have proposed a behavior role-based IDS based on the timing of control signal packets which will be discussed in the next subsection.

4.3.2 Proposed Approach for IDS

We have proposed the network-based ADS which monitors the network packets when the controller sends the control signals to the actuators in the power system as shown in Figure 9. The main notion behind this approach is that the controller provides the corrective actions during specific circumstances like line contingencies/ faults and the frequency of network packets during such events is comparatively low as compared to frequency of packets during the ramp attacks.

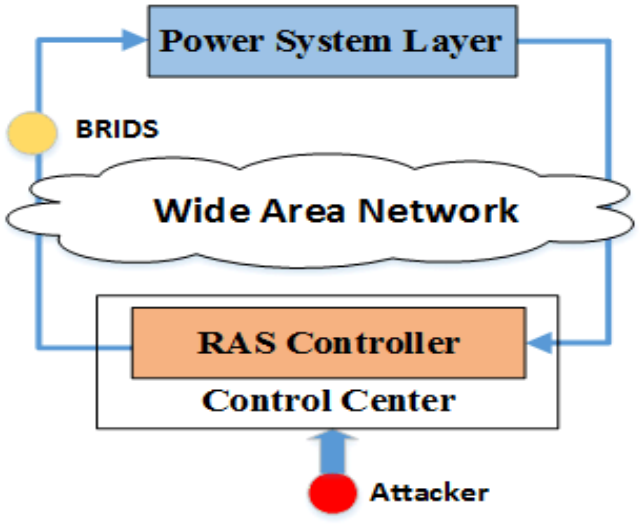


Figure 9. Generic architecture of behavior-rule based ADS (BRADS).

Therefore, based on the in-depth analysis of network packets, we can detect the attacks by assigning the threshold values on the number of control signal packets sent from the controller to the actuators. We have tested our proposed approach in real-time for DNP3 protocol using IDS tools Snort and BRO. Figure 10 shows the proposed approach which can be divided into 5 stages:

1. *Network-packet sniffing*
2. *Protocol packet filtering*
3. *Learning phase*
4. *Rules defining phase*
5. *Real-time detection.*

In stage 1, the network traffic is monitored when the controller is sending control signals to the actuator. In stage 2, the normal DNP3 packet is filtered based on the IP address and port numbers. In stage 3, DNP3 function codes are selected. Finally, in stage 4 and 5, the timing-based rule is defined for the number of function codes based on the packet learning. In stage 4, T_n and T_{n-1}, represent the time of the $n^{th}$and $(n-1)^{th}$ packets where n is the positive integer (n > 0). T_t is the inter-arrival time between the two consecutive packets. We have defined the time threshold, T_{thres}, based on the statistical analysis of the network traffic during the normal disturbances and cyber-

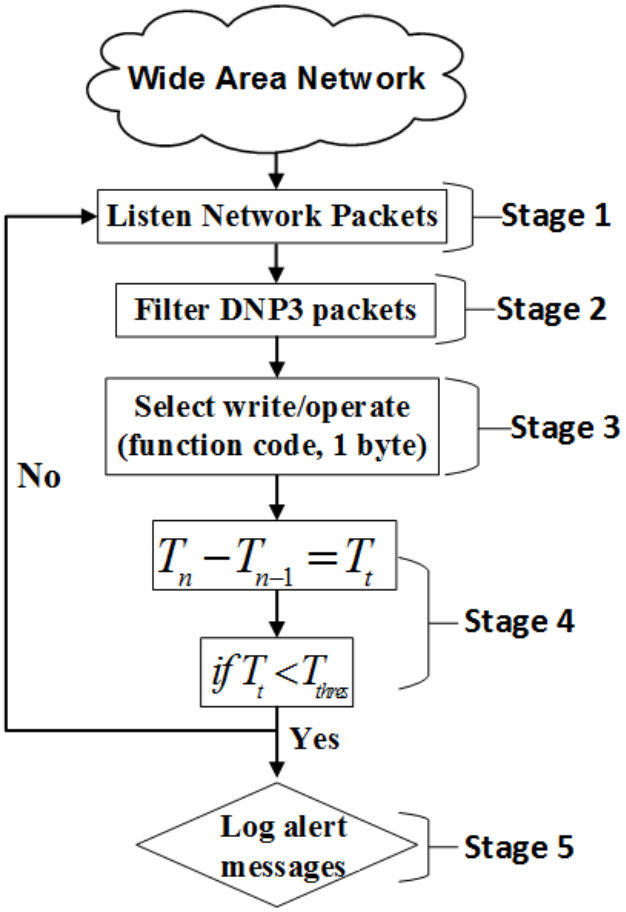


Figure 10. Proposed intrusion detection engine for the ramp attack.

attacks. If the time difference between the two packets, T_t, is less than the defined threshold, T_{thres}, the alert messages are issued to the operator. In this work, we have assigned the value of 0.3 *sec* for two consecutive normal DNP3 packet based on the expert knowledge and literature documents.

4.3.3 IDS Implementation

We have developed and implemented the proposed intrusion detection system by utilizing the network-based IDS tools, Bro and Snort. Figure 11 shows the alerts examples in the log files of Snort and Bro IDS.

> 05/16-19:48:53.33 [**] [1:4444001:1] SCADA_IDS: DNP3 – ramp attack [**] [Priority: 1] {TCP} *.1.200.145:14612 -> *.1.0.38:20000 (**Snort**)
>
> *.1.200.145 did more than 2.0 Function ramp attack! in 0m0s and total is 2 and time is 1495044882.845699 (**Bro**)

Figure 11. Alert examples in the log files of Snort and Bro IDS.

4.3.4 Experimental Setup

Figure 12 shows the experimental setup for the attack-detection experiment using the testbed. We have modeled the modified IEEE 9 bus system on the real time digital simulator (RTDS). The distributed remedial action scheme is implemented in the system where each RASc is operating for a single generator. The controller, RASc2, operating for the generator 2, is communicating

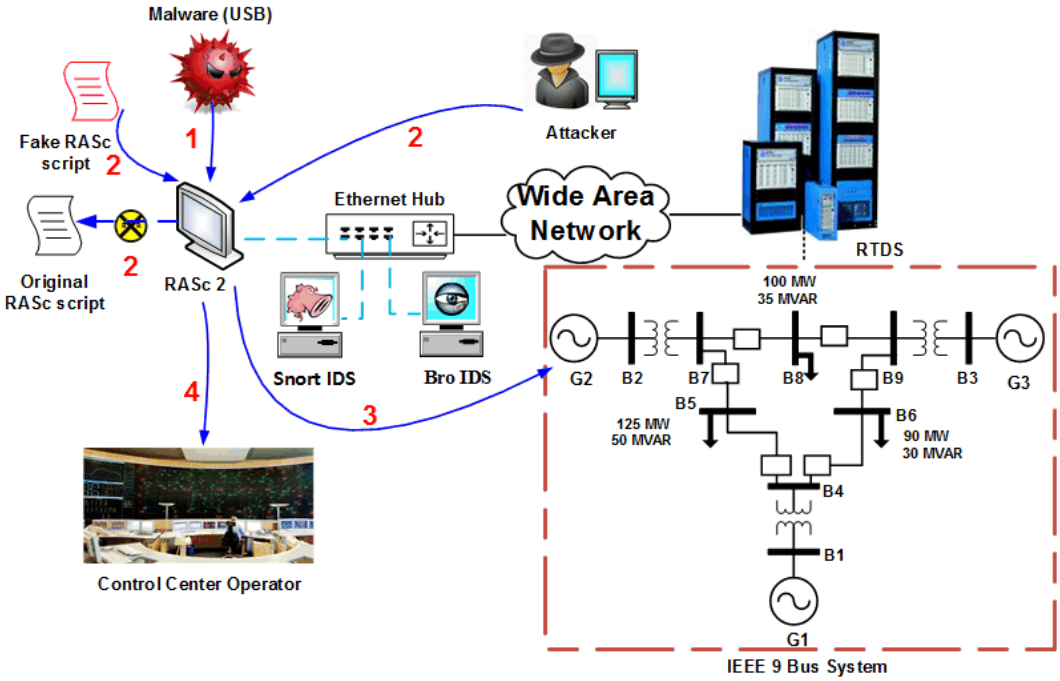


Figure 12. Experimental setup for attack implementation and detection

through the DNP3 protocols to the simulator. It collects data in terms of relay status, line flows and power generation at every 0.125 second and takes corrective actions by shedding different level of generations to avoid thermal overload during the contingency. For simplicity, we have the considered the overhead limit to be 1.5 times of the initial line flows. In the attack scenario, as shown in blue dashed arrows, we have installed the malware (Trojan Horse), written in python script for Windows hosts, in RASc2 which provides unauthorized access to the attacker. Once malware is installed, the attacker transfers the fake RAS script to the affected controller using Cryptcat. The Cryptcat is a Unix utility which allows data/file transferring in encrypted form. In the next step, the attacker closed the original RAS script and malicious script is executed. The malicious script initiates the ramp attack on the generator while sending fake updates to the control center operator. For attack detection, IDS tools Snort and Bro are running in Kali Linux VMware which are listening the ongoing traffic between the controller and RTDS.

4.3.5 Results and Discussions

4.3.5.1 Performance Evaluation of IDS

We have evaluated and compared the performance of Bro and Snort IDS in terms of detection rate and latency in the alert packets. Figure 13 shows the detection rate (a) and the average latency (b)

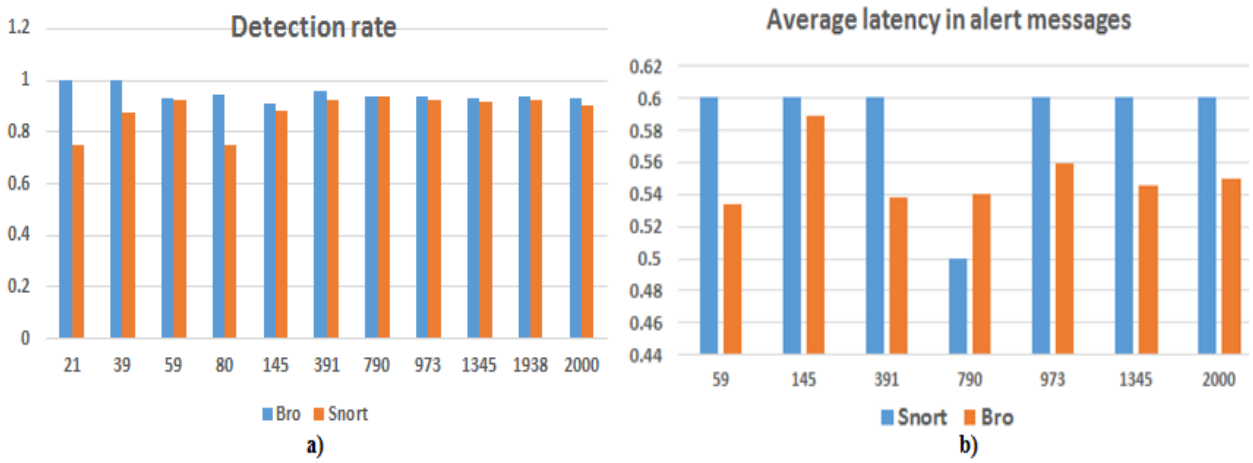


Figure 13. Detection rate (a) and average latency (b) of alert messages for Bro and Snort IDS.

for different sizes of alerts packets. The alert packets are varied from 21 to 2000 packets, and it can be observed that the Bro IDS has performed consistently better with a detection rate greater than 90%. The Snort IDS's detection rate varies sporadically from 93.5% to 75%. For computed average latency, the constant delay of 0.6 s is observed in the most cases of Snort IDS. The Bro IDS has performed slightly faster with the maximum average delay of 0.58 *sec* and minimum of 0.534 *sec*.

4.4 Multi-Agent based Anomaly Detection

In this work, we have proposed the multi-agent based RAS architecture to detect the stealthy coordinated attacks. We have proposed the two-level hierarchical architecture where distributed

local controllers, working as local agents, are operating in different zones/ areas and the overseer, the central agent, receives the local and random measurements from the local controllers and identifies the compromised controller based on the online anomaly detection algorithm. For developing the anomaly detection, local and random measurements are compared, and the measurement errors are computed. Once the measurement error exceeds the defined threshold, the overseer performs the validation checks and the malicious controller is detected based on the two-step verification. We have evaluated its performance through the online testing on the IEEE 39 bus system in the cyber physical environment. More details of this work are provided in the project publication [5].

4.4.1 Coordinated Attack Vectors

We have considered the coordinate attack vectors which include installing the malware on local controller, disabling the original program and running the malicious code, tripping the line maliciously and finally executing the data integrity attacks (pulse, ramp) through the malicious code on the generator while sending false measurements to the operator. Overall, the coordinated attacks involve malware attacks, malicious tripping attacks, generation alteration attack and replay attack in the sequential manner.

4.4.2 Proposed Architecture and Methodology

4.4.2.1 Proposed Multi-Agents based Hierarchical Architecture

Figure 14 shows the proposed MAS based hierarchical architecture for the distributed RAS scheme, where each controller is operating as a local agent at the substation, and the overseer, the central agent is periodically monitoring the local agents while operating at the control center. The local agent is working as a substation-based protection controller which is responsible for monitoring and protection of the associated zone independently. It collects information from the local sensors (PMUs, relays) deployed in the specific zone and performs corrective actions through the actuators (MW, MVAR control) to mitigate different power disturbances. Apart from the local state information, each local agent is also collecting measurements from the other zonal sensors as shown in the figure to introduce the redundancy in the system through the client-server communication. Each local controller forwards the collected local measurements to the overseer along with the other zone measurements in the dynamic, random manner. The overseer collects the local and random measurements for the system states and detects anomalies using the proposed data-driven anomaly detection algorithm which we have discussed in the next subsection.

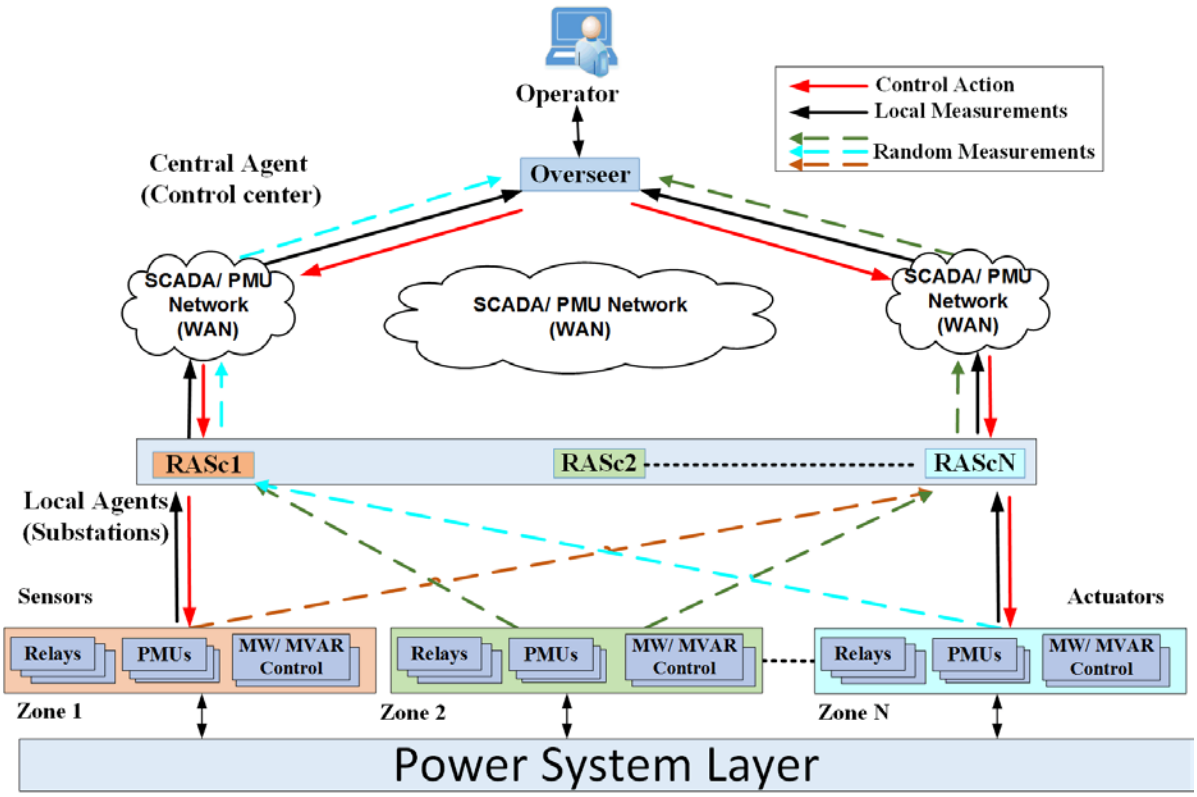


Figure 14. Overview of the proposed architecture for multi-agent RAS.

Table 3. Roles assigned to different agents

Roles assigned	***Agents***
Receive updates from RAScs Send Check commands to RASc Alert the operator during cyber attacks	overseer (Central)
Receive System measurements Take corrective actions Send updates to overseer	RASc (local)

We have proposed the two-level hierarchical architecture where central and local agents are performing their separate respective functions as shown in the action

Table 3. We have assigned three different roles/ functions to the overseer, which include receives updates from the local RAScs, sends Check command to the RASc during the measurement error, and alerts the operator whenever the cyber-attacks are detected. The three roles assigned to the local RASc are receiving system measurements in terms of relays status, the line flows, generator output, taking corrective actions as needed during the disturbances, and sending periodic updates to the overseer with local and dynamically changing random measurements.

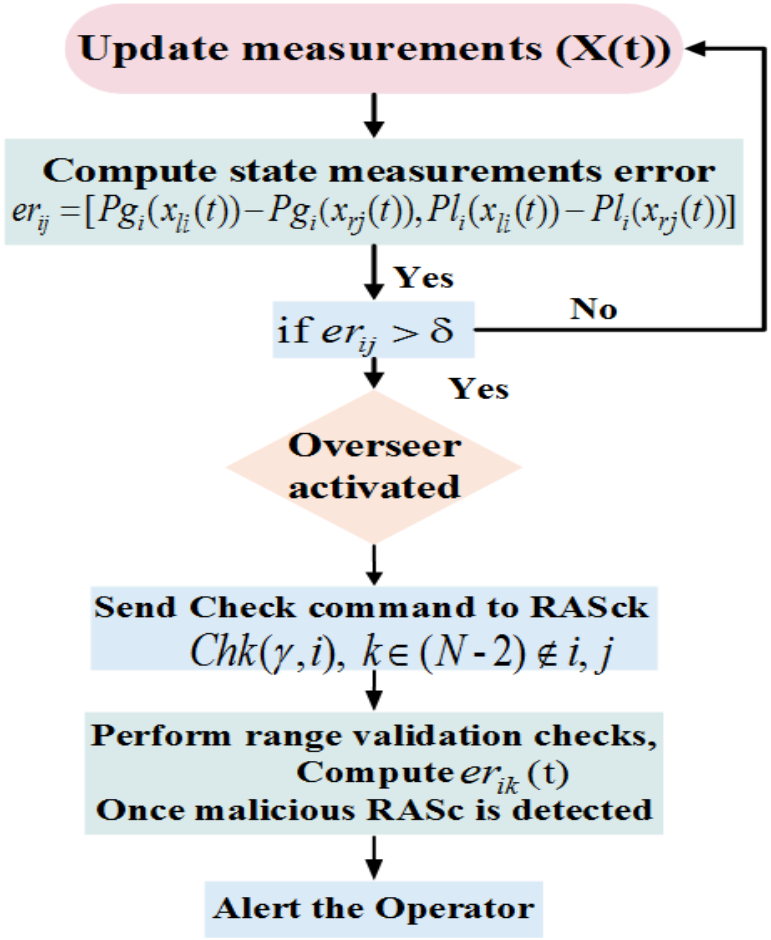


Figure 15. Anomaly detection methodology to detect the compromised RASc.

4.4.2.2 Proposed Anomaly Detection Algorithm

Figure 15 shows the proposed anomaly detection methodology running at the overseer. The overseer periodically updates the measurements coming from the local agents and computes the state measurement error, $er_{ij}(t)$, of the generation, (Pg_i), and line flows, $(Pl_i(t))$, from the i^{th} zone, $[Pg_i(x_{li}(t)),\ Pl_i(x_{li}(t))]$ and j^{th} zone, $[Pg_i(x_{rj}(t)), Pl_i(x_{rj}(t))]$ at a particular instant t. For the computed error, $er_{ij}(t)$, i and j represents the local and outside zones at time t. The parameter δ is defined as the error threshold value. When the computed error exceeds the defined threshold, check command, $Chk(\gamma, i)$, is sent to the third controller, RASck, where k is selected randomly from the remaining N-2 controllers which are not involved in the error conflict. It is required to perform the range validation checks based on the error computed from the kth controller's measurement of the ith zone with local i^{th} controller's (RASci) measurements, as defined by $er_{ik}(t)$. If the computed error, $er_{ik}(t)$, exceeds the given threshold, the overseer declares that the RASCi is compromised and alert is sent to the operator. It is important to note that the overseer is performing two-step verification to successfully detect the compromised RASc.

4.4.3 Experiment Setup and Case Studies

Figure 16 shows hardware-in-the-loop based experimental set-up for the attack implementation. We have modeled the modified IEEE 39 bus in ePHASORsim and simulated in the OPAL-RT, real time digital simulator. Figure 17 shows the topology of the system and it is divided into three different zones or areas where decentralized RAS is implemented for each zone to prevent the thermal overloading during line outages. We have implemented the coordinated attacks on the RASc2 which is collecting measurements from the simulator. The simulator is integrated with two physical relays, as representing line L16-21, L16-24, which are connected to the remote terminal unit (RTU). For successful attack completion, we install the malware, Trojan Horse, written in python script (step1), which provides the backdoor connection to the attacker's computer (step 1).

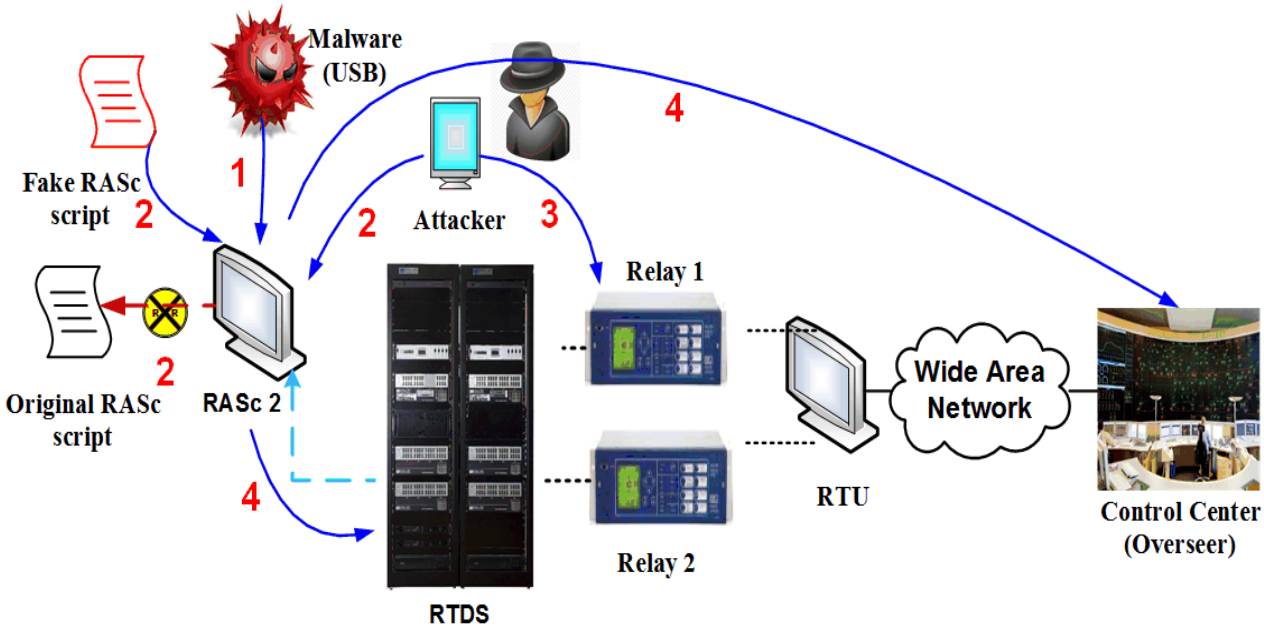


Figure 16. Experimantal set-up for attack implementation in PowerCyber Lab.

Next, we close the python program running for legitimate RASc and execute the python program for malicious RAS (step 2). Afterwards, the malicious tripping attack is performed by replaying the captured tripping packet on relay 1 which disconnects the line 16-21 to trigger the RAS (step 3). Finally, the attacker initiates the pulse/ ramp attack on the generator 35, while sending false measurement updates of generation to the overseer, running at the control center to hide the malicious action (step 4). Once the attack is successfully performed, we have collected the system data with timestamps from the simulator which is used later for the detection testing.

Figure 18 shows the MAS based online anomaly detection topology for 3 zones/ areas decentralized RAS where, each controller receives the list of local and random measurements. Each RASc is centrally monitored by the overseer. The RASc2 is operating in the zone 2, which polls the local zone measurements in terms of generator, G35, and line status, L16-21 and L16-24. It also polls the outside zone's measurements from zone 1 and zone 3 in terms of generators, G38, G32, and line statuses, (L26-29, L26-28), (L6-11, L13-L14) as shown in the figure.

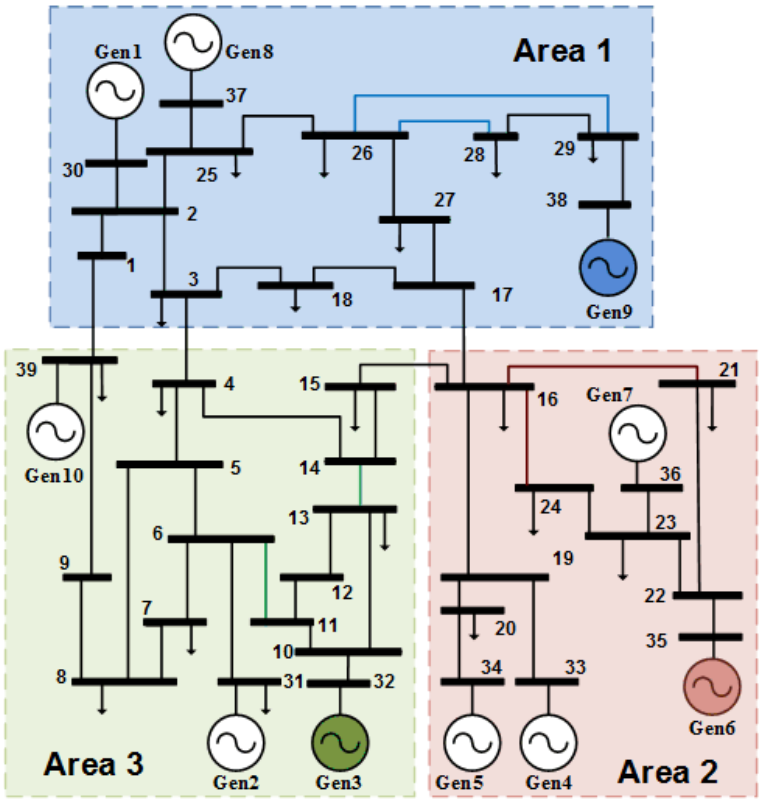


Figure 17. Decentralized RAS enabled modified IEEE 39 bus system.

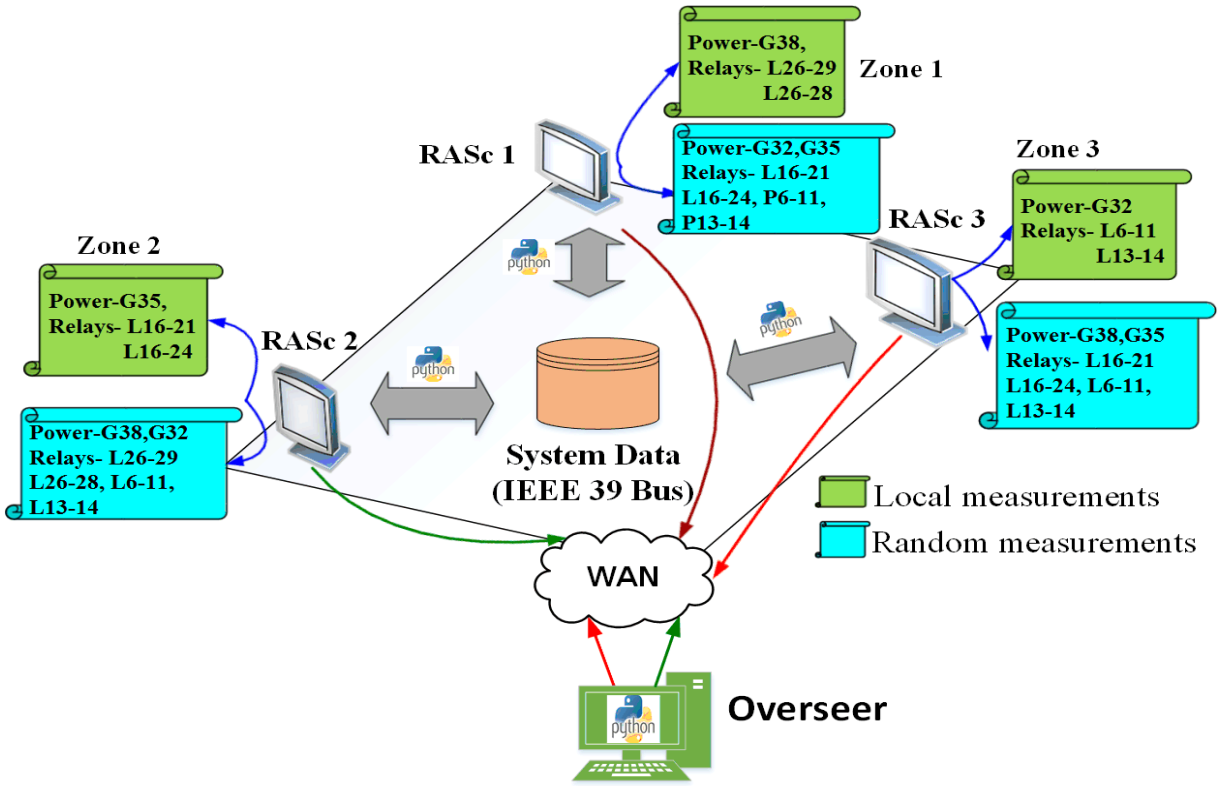


Figure 18. Online anomaly detection topology for the 3 zones RAS.

For the detection testing, we develop the MAS using python script where each RASc is sending their measurement updates every 4 second to the overseer. Initially, the system data with timestamps is stored in the python script of each controller, which initiates interaction with the overseer at the same time. The overseer is processing their data to the online anomaly detection algorithm which is also running in the python script. Once the anomaly is detected, an alert is issued to the operator.

4.4.4 Results and Discussions

4.4.4.1 Performance Evaluation

Figure 19 (a) and (b) show the online performance of the proposed algorithm. The proposed algorithm is evaluated in terms of detection and latency cycles during the pulse and ramp attacks. It can be observed that pulse attack is detected faster than the ramp attack because the pulse attack causes the sharp deviation from the initial state as compared to the ramp attack for the large detection threshold. For the small detection threshold, number of cycles to detect both attacks are almost same. For the average latency, it can be observed that we are able to successfully detect both attacks with an average delay of 0.389 and a maximum delay of 1.25 cycles.

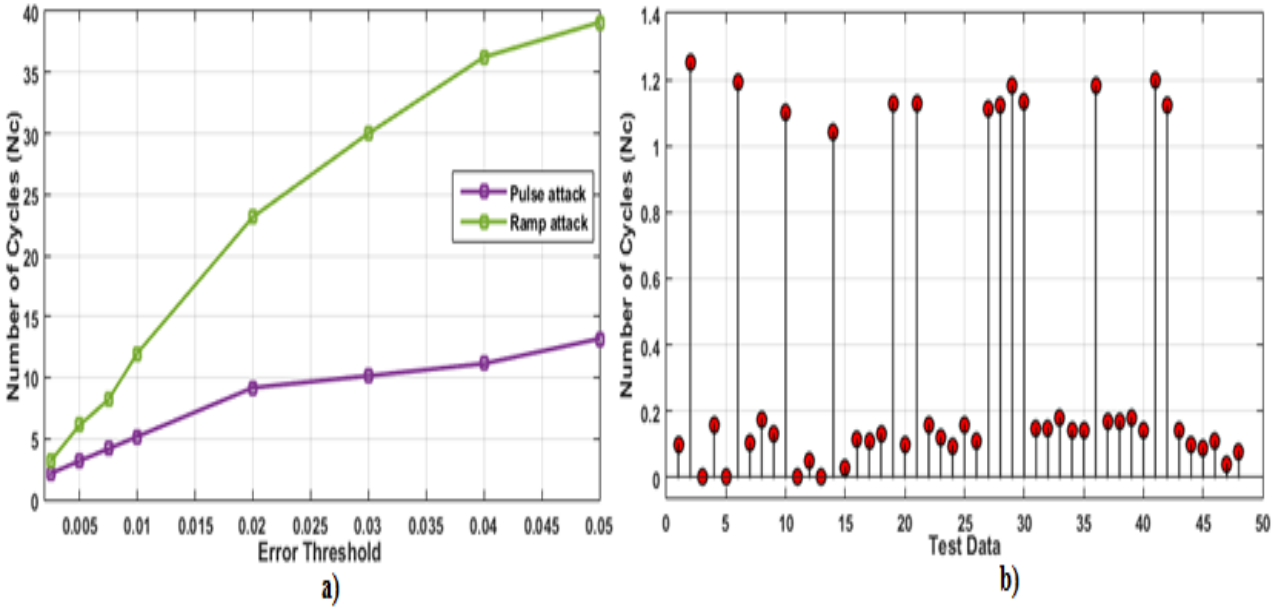


Figure 19. Detection Cycles (a) and Latecy cycles (b) for pulse and ramp attack.

5. Anomaly Detection for Automatic Generation Control

5.1 Basics of AGC

AGC is a wide area control operation that is used for maintaining the system nominal frequency by balancing the generation with the load and limiting the tie line flows between the balancing areas. We have considered two types of AGC schemes: Conventional AGC, and Proportional Integral Derivative (PID) AGC. Both schemes employ Area Control Error (ACE) calculated from the frequency and tie line flow deviations every 2-8 seconds using equation 5.1, where, ΔP_{tie} represents the difference in the tie line flows from the actual and schedule tie-line power flows, Δf represents difference between the actual frequency, f_{act}, and nominal frequency, f_{nom}, and β is the balancing authority bias.

$$ACE = \Delta P_{tie} + \beta \Delta f \tag{5.1}$$

5.1.1 Conventional AGC

This is the traditional method of AGC in which the ACE values (E), are consecutively added to generate the control signal. The control input, *C(t),* at time 't' is given by equation 5.2.

$$C(t) = E_0 + E_1 + ... + E_t \tag{5.2}$$

5.1.2 Proportional Integral Derivative (PID) ACE based AGC

The traditional method can be rewritten in the PID form as a discrete PI type controller with both K_p and K_i parameters are set to 1. We can consider the current ACE as the proportional component and the sum of previous errors as the integral component.

$$C(t) = 1 * E_t + 1 * (E_0 + E_1 ..... + E_{t-1} \tag{5.3}$$

By using variable parameters and including a derivative component consisting of the difference between the current and its previous error, a PID form of AGC can be constructed as shown in equation 5.4.

$$C(t) = K_p * E_t + K_i * (E_0 + E_1 ..... + E_{t-1}) + K_d * (E_t - E_{t-1}) \tag{5.4}$$

5.2 Model based Anomaly Detection

We have proposed the model-based anomaly detection and mitigation algorithms for the AGC operation while considering different percentages of renewable integrated with the bulk power system. In this work, we have deployed two types of AGC: Conventional AGC and Proportional Integral Derivative (PID) based AGC. The ramp attack is implemented on both types of AGC with various levels of renewable penetration. Finally, the attack detection and mitigation algorithms are tested in various scenarios to evaluate the ADS performance and comprehend the impact of the mitigation algorithm on the system during the normal and N-K contingencies. The proposed

algorithms are tested in the experimental environment using the PowerCyber CPS security testbed at Iowa State University. More details of this work are provided in the project publication [6].

5.2.1 Cyber Attack Vector

5.2.1.1 Ramp Attack

During the ramp attack, a constant ACE value is injected to the generator, which causes the constant increase or decrease of the generator output and, eventually leads to a constant rise or drop in the system frequency. The equation 5.5 shows the generator control signal after the attack has started, where, $C(t)$ is the control signal before attack, E_a is the attack magnitude, and t_a is the attack duration.

$$C(t+t_a) = C(t) \ + E_a * t_a \tag{5.5}$$

5.2.2 Attack Detection and Mitigation

This subsection discusses about the attack detection and mitigation algorithm that can detect the attack at the initial stage and take proper mitigation to prevent the attack from affecting the system beyond the certain limits. Figure shows the flow chart for the proposed algorithm. Once the ACE and frequency values are obtained for each cycle, two conditions are checked:

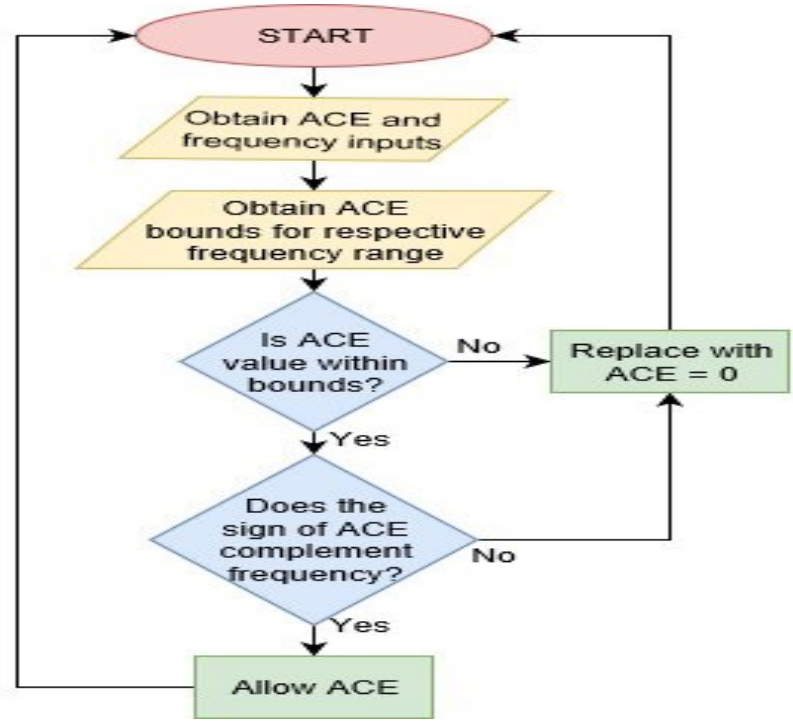


Figure 20. Algorithm for attack detection and mitigation.

1. The values are checked to be within the pair of bounds.
2. The sign of ACE value is checked with the respective frequency value to ensure that the computed ACE value is utilized to improve the frequency conditions.

In case of violation of either condition, the ACE value is dropped and substituted with null (0). By this a ramp attack would be stopped immediately during the excursive ACE values as defined by condition 1, or after the first few cycles when condition 2 would not be satisfied, if the attack value was within limits. We have determined the ACE bounds by leveraging the data used by regulators for determining the performance parameters of the AGC (viz. Control Performance Standards - CPS1 and CPS2) and the Balancing Authority ACE Limits (BAAL) [19]. The data contains one-minute average values of frequency and ACE for a total period of 12 months which is updated every month. We have considered the bounds for each frequency range by selecting the largest ACE values observed in that range in the past 12 months. Figure 21 shows the proposed methodology for computing the ACE bounds. It assumes that the generating stations have access only to frequency values and tie-line flow values are not used as a secondary parameter.

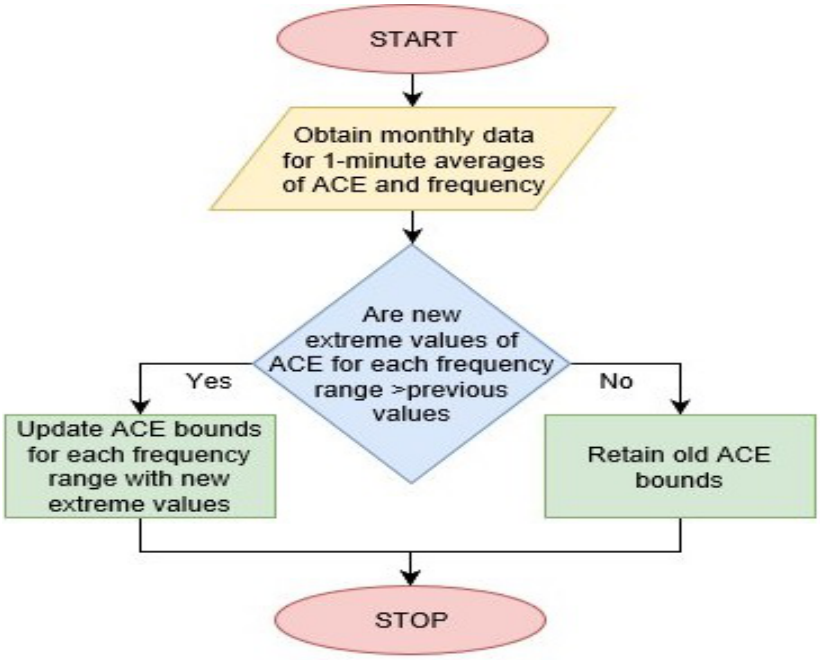


Figure 21. Procedure to determine ACE bounds for mitigation.

5.2.3 Experimental Implementation

We have performed the experiments on the modified IEEE 39 bus system, simulated in OPAL- RT. Figure 22 shows the system which is divided into two Balancing Authorities (BAs). Generators 1 & 8, and 2 & 3 are involved in the AGC operation in BA1 and BA2 respectively. We have modified the model to simulate renewable energy sources. Generators 4, 5 and 9 are converted to renewable plants successively for simulating the increasing penetration of distributed sources. In order to simulate the variation in the wind turbine output, the individual machines in the plants were simulated with an input value subjected to 1% fluctuation.

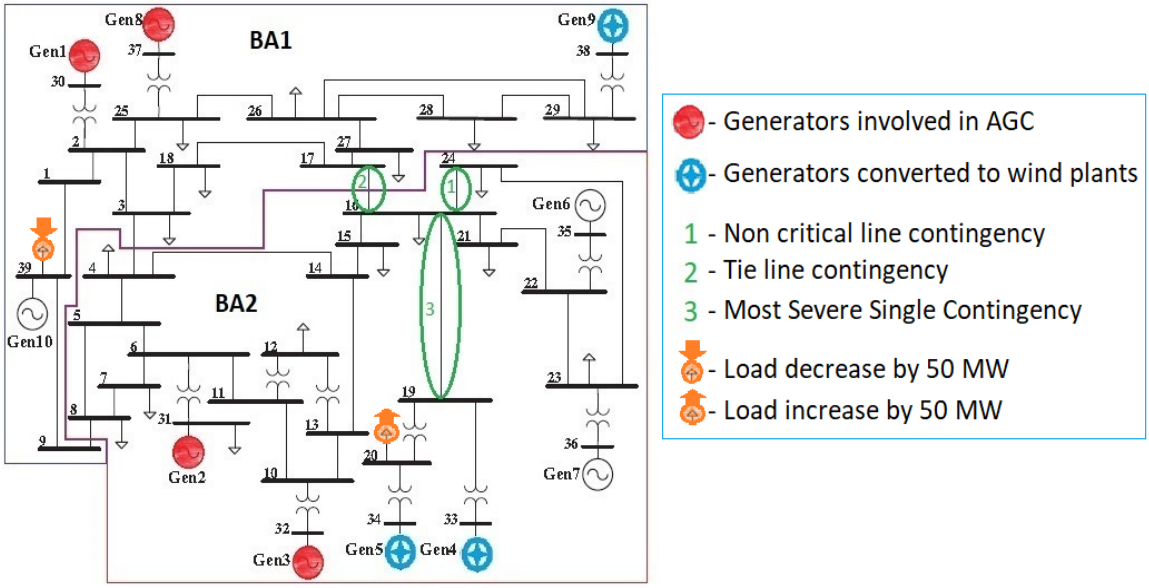


Figure 22. IEEE 39 bus model.

5.2.4 Performance Evaluation

5.2.4.1 Attack Analysis

The ramp attack was conducted on two cases - attack on one ACE and two ACE values, for four different conditions of renewable penetration 0%, 10%, 20% and 30%. The increase in renewable penetration has two consequences on the grid:

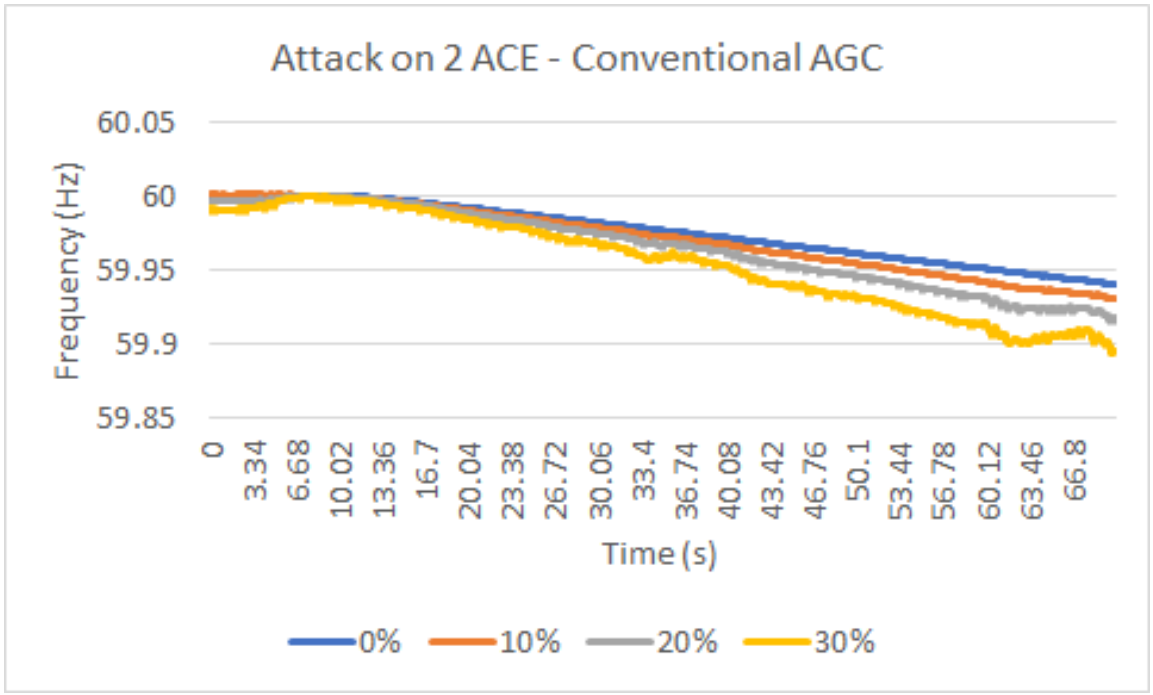


Figure 23. Attack comparison for different levels of renewables.

1. Fluctuating output power leading to frequency fluctuations
2. Reduction in system inertia leading to faster frequency response and higher rate of change of frequency (ROCOF)

The ACE value was attacked and replaced with -5 MW for a duration of 1 minute (60 seconds or 15 AGC cycles) causing a drop of 75 MW of generation. When both the ACE signals are attacked there is a drop of 150 MW. From the plot shown in Figure 23, it can be inferred that with increase in renewables, due to reduction of inertia, the ROCOF is higher and the drop in frequency due to the attack is faster. The attack was then repeated for the same scenarios with PID based AGC. Due to the reduced control signal used by this algorithm, by the end of 2 minutes the reduction in generation was only 40 MW for attack on 1 ACE, and 80 MW for attack on 2 ACE values. This results in lesser frequency drop as shown in Figure 24. Thus, it can be inferred that attack on PID based AGC has lesser impact. The results obtained during the attack analysis are summarized in

Table 4.

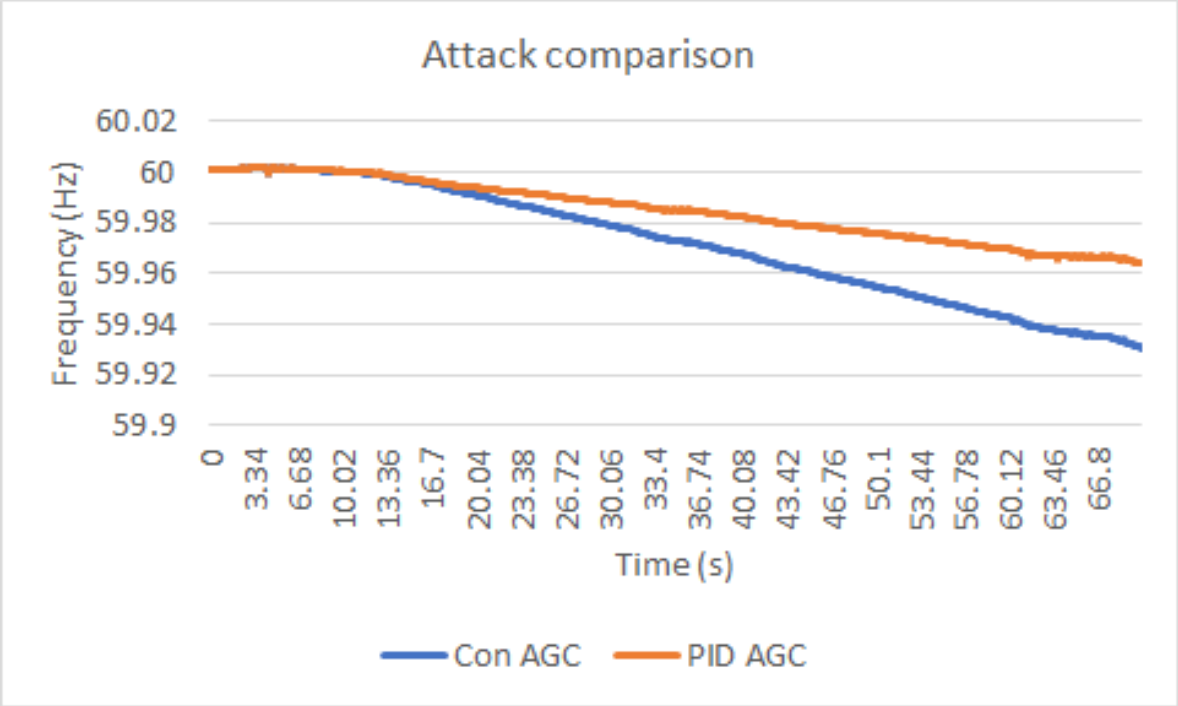


Figure 24. Attack comparison between Conventional & PID based AGC.

Furthermore, the study was also performed on the performance of the PID based AGC as compared to the conventional AGC. Based on a two minutes observation of the two algorithms for the four cases of renewable penetration, it was observed that for the considered system conditions,

Table 4. Attack impact analysis

Renewable Penetration (%)	**Frequency Drop (Hz)**			
	Attack on 1 ACE		**Attack on 2 ACE**	
	Conventional AGC	PID based AGC	Conventional AGC	PID based AGC
	Generation drop = 75 MW	Generation drop = 40 MW	Generation drop = 150 MW	Generation drop = 80 MW
0	0.03	0.016	0.059	0.033
10	0.036	0.016	0.07	0.036
20	0.044	0.016	0.083	0.044
30	0.055	0.028	0.106	0.06

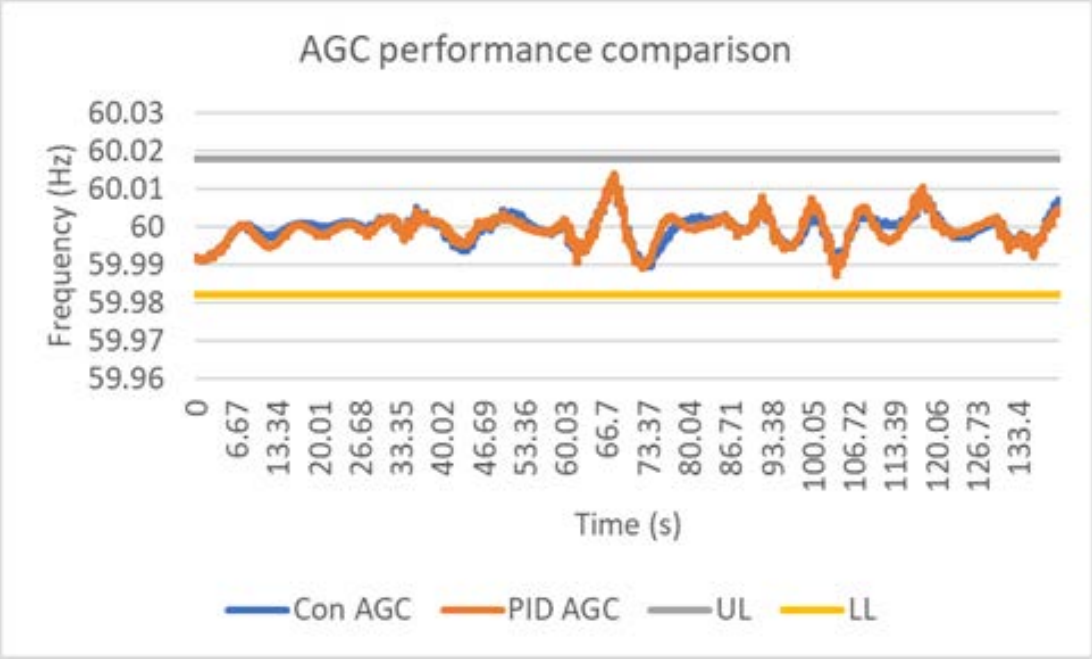


Figure 25. Performance comparison between Conventional & PID based AGC.

PID base AGC algorithm provides a satisfactory performance that is as good as the conventional algorithm, since it manages to maintain the steady state frequency error within the frequency regulation requirements (18 mHz) as shown in the Figure 25.

5.2.4.2 Defense Analysis

The ramp attack was conducted for the same attack magnitude and duration with the mitigation algorithm in place for both the AGC methods, with varying conditions of renewables, and with attacks on single and multiple ACE values. It was observed that the attack was prevented during the AGC cycles for which condition 2 was not satisfied. Condition 1 was not violated as the attack magnitude was small. This resulted in the possibility of the attack being rarely successful and the successful ones having negligible impact. Figure 26 shows the impact of the attack on both ACE values for conventional AGC with the mitigation implemented.

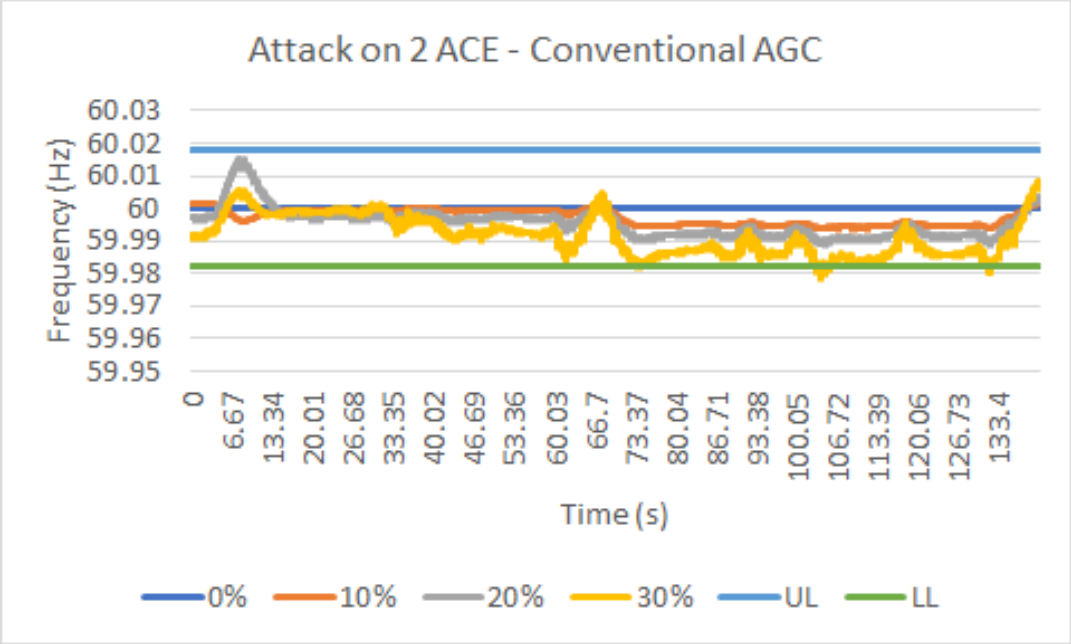


Figure 26. Effect of mitigation algorithm on attack.

5.2.4.3 Effect of Mitigation on AGC Performance

From the algorithm, it is evident that an alert would be generated whenever the ACE signals have opposite signs. This will result in a high incidence of false positives as this condition is observed most of the time. So, it is important to ensure that the normal AGC operation is not adversely affected by the mitigation. The AGC operation was simulated with the mitigation in place for both AGC algorithms and different conditions of renewables. Due to frequency being the primary factor in the algorithm, the system tends to restore frequency within the permissible errors of 18 mHz without any hindrance, despite the false positives. To further validate AGC performance, an analysis was conducted by means of a step change in two of the loads. One load in BA1 and one in BA2 were subject to an increase and decrease of 50 MW respectively as shown in Figure 27. In this case, both the BAs would need to achieve a ramp up and ramp down in their generation by an equal amount, while being allowed to do only one at a time and needing longer time for restoration. It was observed that the system performance was satisfactory with the mitigation algorithm in place. The upper plot in Figure 27 shows the frequency restoration. Even though the frequency performance appears to be inferior, there is a very less possibility of violation. This seemingly

inferior performance is because at any instant, the machines in the grid are allowed to either ramp up or down, but not both. However, because of condition 2 in the mitigation algorithm, the frequency deviation in any direction will always be facing an aggressive opposition. This would cause relatively greater frequency swings as compared to the system without mitigation. From the lower plot, it was observed that the tie line restoration occurs at fairly the same duration with or without mitigation. In general, it shows the performance of detection and mitigation algorithm. It is observed that the system performance was satisfactory with the algorithm in place. The upper plot in Figure 27 shows the frequency restoration during the ramp attack. The system frequency restores quickly, which has less possibility of violations. The lower plot shows that the tie line restoration occurs at fairly the same duration with or without mitigation.

Figure 27. AGC performance with mitigation algorithm.

5.3 Machine Learning based Anomaly Detection

We have proposed the machine learning based ADS for abnormal generation controls induced by different single cyber-attacks. Specifically, we have proposed the semi-supervised clustering algorithm with Hierarchical Density based Spatial Clustering of Application with Noise (HDBSCAN) for ADS against the generation control under ramp attack, switching attack, AGC integrity attacks, etc. We have evaluated the proposed algorithm through the experimental setup and shows that the proposed algorithm provides better detection accuracy than K-means clustering.

Furthermore, it can also provide detailed classifications of the normal and abnormal generation controls as well as among the various anomaly scenarios. More details of this work are provided in the project publication [8].

5.3.1 Abnormal Generation Control Detection

1.3.1.2 Overall Anomaly Detection Process

Figure 28 shows the overall process of the proposed ADS. Initially, we have collected the sample data which is processed through the metric calculation module, and then a tuple of 3 metrics is formed as one data instance. The three metrics compute the data instances as input features which are sent to the on-line detection model to detect any abnormal generation control. We are also storing the data instances in the database. The proposed Semi-supervised clustering will be carried out off-line on request to complete the training process and, on-line model will be updated accordingly.

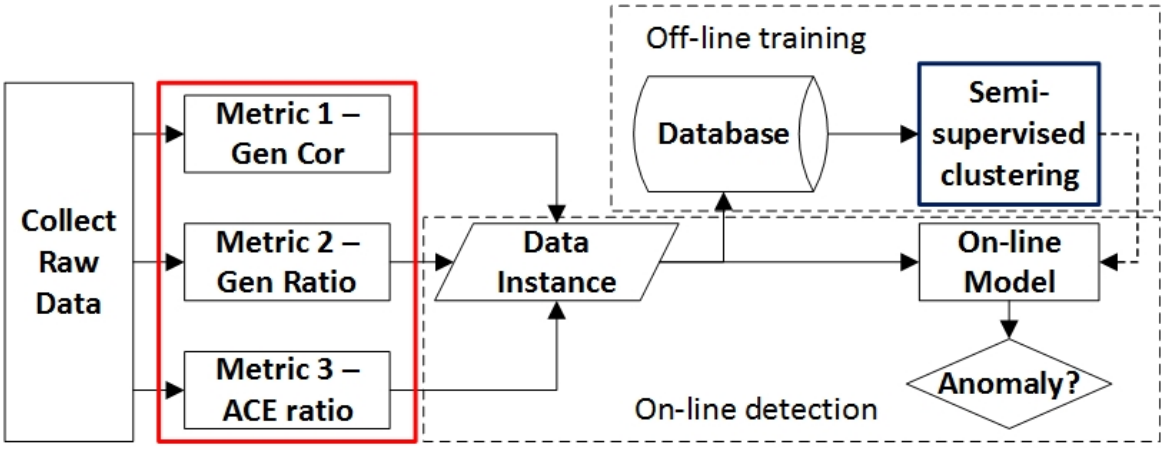


Figure 28. Generation control just before AGC operation.

5.3.2 Semi-Supervised Clustering with HDBSCAN

Reference [20] provides the detailed discussion about HDBSCAN for semi-supervised clustering. It shows how to utilize the labeled data instances as constraints at the instance level to find the optimal clusters, especially to list the pairs of points that should stay in the same cluster. When the pairs of points are not in the same cluster, clusters with the least constraint violations are selected. In this work, we have come up with a clustering methodology as depicted in the Figure 29. The proposed algorithm recursively finds the clusters in the manner of divide and conquer.

Algorithm 1 Div_Conq_HDBSCAN()

Input: $\mathbf{X}_{N\times p}$, $C_{N\times 1}$, $\{m_{pts}\}$, and T_{depth}
Result: $C_{N\times 1}$

procedure DIV_CONQ_HDBSCAN
  Pick up one element m from $\{m_{pts}\}$
  Run HDBSCAN($\mathbf{X}$,m) and attain cluster tree cl
  If Bad cluster tree pattern observed in cl
    Divide $\mathbf{X}$ and proceed to next layer of recursion
  else
    Find all true splits, cuts at depth above T_{depth}
    Select the cut with min sum of cluster entropy
    go to step 2 until $\{m_{pts}\}$ becomes empty
  end if
  Modify $C_{N\times 1}$ according to the optimal cl
  If Clusters with high entropy still exist
    Proceed to next layer of recursion
  end if
  return $C_{N\times 1}$
end procedure

Figure 29. Proposed algorithm for semi-supervised clustering.

5.3.3 Experimental Results

5.3.3.1 Introduction of the Datasets

Initially, we have collected the synthetic dataset based on the IEEE 39 bus system [20]. The system is divided into 3 BAs as shown in Figure 30. The BA2 (Area 2) is the main area under investigation. The generator 4, G4, in BA2 is modeled as a primary control unit and the generator 5, G5 and generator 6, G6, as secondary control units. We have only considered G5 as the main target for the cyber-attacks. We have simulated six different scenarios that include 1) normal events occur inside of BA (nor in), 2) normal events outside BA (nor out), 3) modify G5 ACEs to negative values (flip attack), 4) modify G5 ACEs to a wrong constant (constant or scaling attack), 5) keep ramping a generation unit up or down after an intrusion (ramp attack), 6) switch between two different generation levels continuously after an intrusion (switching attack). The dataset is summarized in Table 5, and, also depicted in Figure 31.

Table 5. Training and testing datasets for AGC

Cases	***Training***		***Testing***
	Labeled	Unlabeled	
Nor in	102	-	73
Nor out	91	-	23
Flip attack	279	-	13
Constant attack	275	-	20
Ramp attack	12	-	12
Switching attack	14	-	16
Total	773	470	157

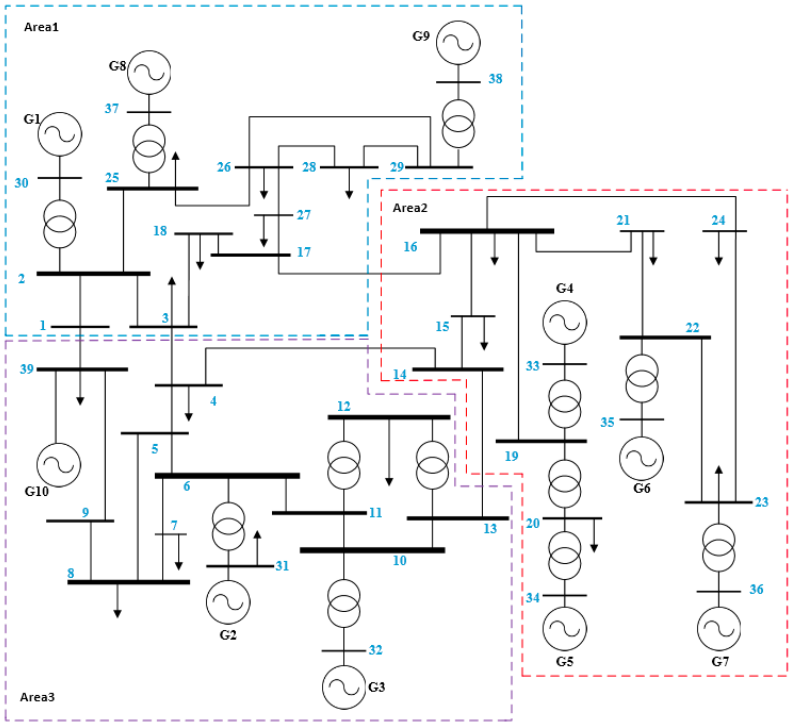


Figure 30. IEEE 39 bus model divided into 3 Bas.

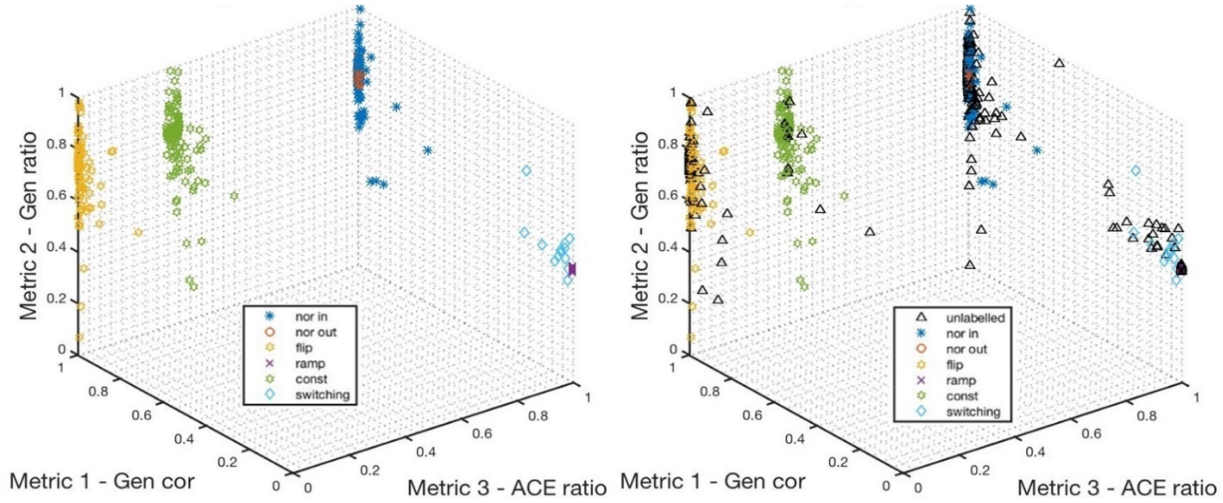


Figure 31. Training datasets.

5.3.3.2 Clustering with K-means

K-means clustering was applied in [21]. The result is simply echoed here as a baseline for the new clustering algorithm. Figure 32 shows the final clusters obtained, and Table. II of Figure 34 is the confusion matrix for the test.

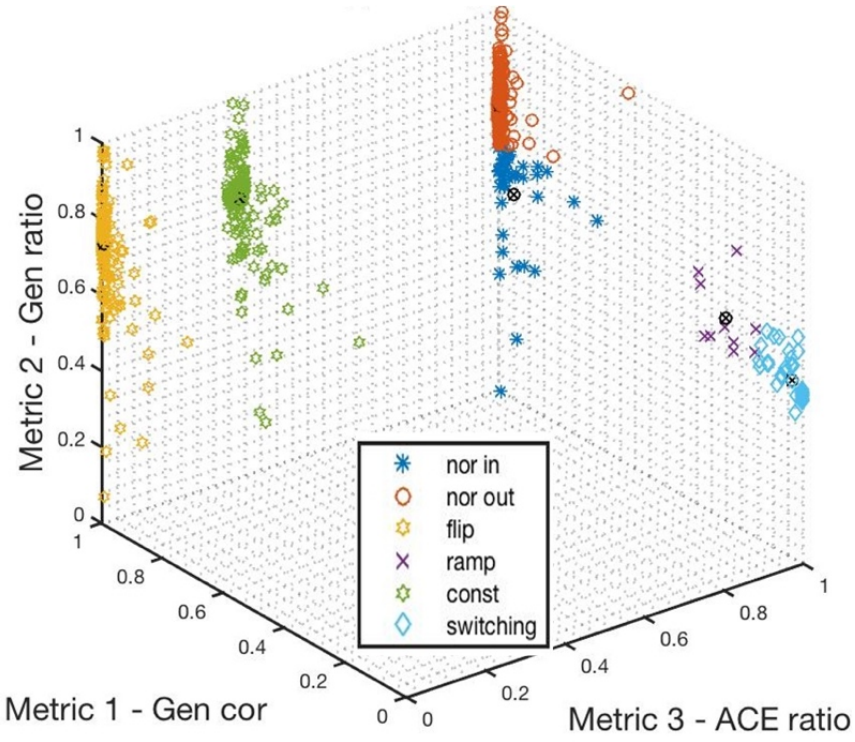


Figure 32. Clustering results with K-means.

5.3.3.3 Clustering with Div_Conq_HDBSCAN

Figure 33 illustrates the performance of the proposed clustering methodology. It shows the process of the first recursion of the semi-supervised clustering, and different clusters are properly separated, which can be clearly observed in the figure. Clustering with HDBSCAN also provides the labeling of potential outliers present in the dataset, which should be further investigated by experts. We have also performed the online detection, where, KNN is applied to the testing dataset and results are summarized in Table II of Figure 34. The overall misclassification rate is lower in the proposed methodology as compared to the KNN based clustering as shown in Table III of Figure 34.

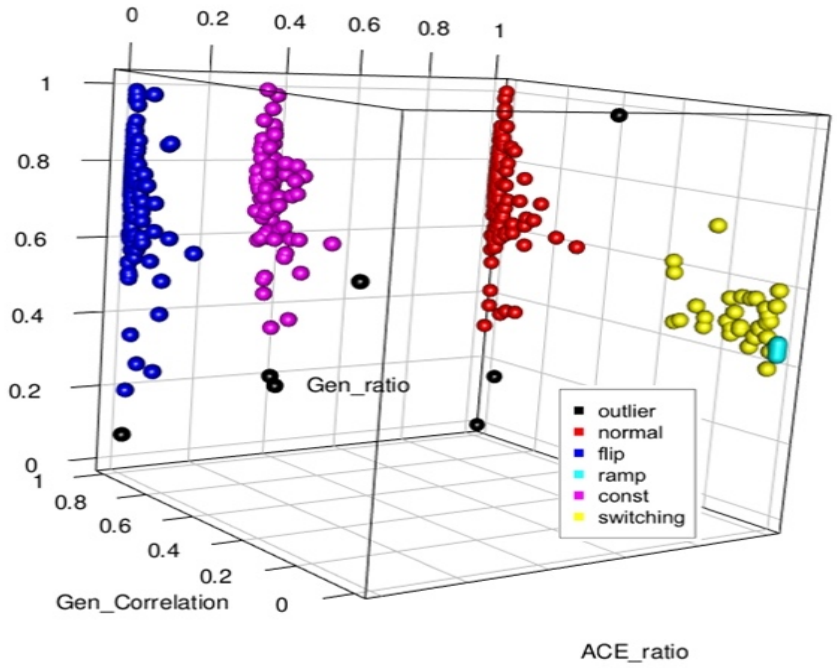


Figure 33. Clustering results with HDBSCAN.

TABLE II
TEST CONFUSION MATRIX OF K-MEANS CLUSTERING

	Prediction (%)					
	nor in	nor out	flip	const	ramp	switching
nor in	31.51	67.12	0	1.37	0	0
nor out	0	100	0	0	0	0
flip	0	0	100	0	0	0
const	0	0	0	100	0	0
ramp	0	0	0	0	0	100
switching	0	0	0	0	6.25	93.75

a)

TABLE III
TEST CONFUSION MATRIX OF PROPOSED CLUSTERING METHODOLOGY

	Prediction (%)				
	normal	flip	const	ramp	switching
normal	100	0	0	0	0
flip	0	100	0	0	0
const	0	0	100	0	0
ramp	0	0	0	100	0
switching	0	0	0	18.75	81.25

b)

Figure 34. Test confusion matrix of K-Means Clustering (TABLE II) and proposed clustering (TABLE III).

6. Conclusions

In this project, the problem of detecting different types of attacks is addressed by developing sophistical anomaly detection systems using the machine learning, temporal behavior based, historical data correlation and multi-agent-based architectures and methodologies. Based on the multitude of vulnerabilities for the existing WAPAC architecture, we have investigated different types of attacks. We have described several steps involved in creating and implementing the attacks using the experimental setup available at ISU's PowerCyber testbed. Specifically, we have considered single attack vectors which include malicious tripping attack, malware attack, generation altering attacks (ramp/pulse) as well as multiple attacks operating in a coordinated fashion.

For Wide-Area Protection scheme (WAP), also known as remedial action scheme (RAS), we have proposed three different types of anomaly detection systems (ADS): 1) Machine learning based ADS, 2) Temporal behavior-based ADS, and 3) Multi-agent based ADS. In machine learning based ADS, we showed how the phasor measurement units can be leveraged to develop the decision tree-based detection rules for detecting the malicious tripping attack and distinguishing it from the normal line tripping during the power system disturbances. We have implemented the attacks and then performed the offline and real-time testing in a cyber-physical environment. In temporal behavior-based ADS, we have showcased the application of Intrusion Detection System (IDS) tools, Snort and Bro, for detecting the generation altering attacks. The detection approach was developed for DNP3 protocol, however, it can also be applied to other SCADA based protocols. In multi-agent-based ADS, we propose a two-level hierarchical multi-agent based architecture which consists of distributed local agents which are periodically monitored by the overseer, the central agent. We have proposed the anomaly detection methodology based on random measurement updates, inspired by MTD based strategy, to detect the stealthy coordinated attacks.

For Wide-Area Control, also known as automatic generation control (AGC), we have discussed two types of anomaly detection system: 1) Historical data-based ADS, 2) Machine learning based ADS. The main notion of historical data-based ADS is to introduce the redundancy which can improve the observability of the system and thus, it helps in detecting the cyber-attacks. In historical data-based ADS, we have employed the historical monthly data for creating different bounds of ACE for the given frequency range, and during the online testing, we have checked the ACE values to detect the anomalies. In machine learning based ADS, we have proposed the semi-supervised clustering algorithm with Hierarchical Density based Spatial Clustering of Application with Noise (HDBSCAN) for detecting different types of attacks. It utilizes cluster entropy to select the optimal cut of a cluster dendrogram which helps in the clustering process. Experimental results have demonstrated that it has better accuracy than K-means algorithm in identifying different types of attacks.

References

[1] A. Ashok, M. Govindarasu, and J. Wang, “Cyber-Physical Attack-Resilient Wide-Area Monitoring, Protection, and Control for the Power Grid,” Proceedings of the IEEE, Vol. 105, no. 7, pp. 1389-1407, 2017.

[2] NERC Critical Infrastructure Protection Committee (CIPC), “Cyber Attack Task Force (CATF) Update,” North American Electric Reliability Corporation (NERC), Dec. 2011.

[3] ICS-CERT,”Monitor (ICS-MM201212),” January 2012 [Online].

[4] B. Miller and D. Rowe, “A survey of SCADA and critical infrastructure incidents,” Proceedings of the First Annual Conference on Research in Information Technology, pp. 51–56, 2012.

[5] N. Falliere, L. O’Murchu, and E. Chien, “W32.stuxnet dossier,” Technical report, Symantec, Feb. 2011.

[6] ICS-CERT, “Cyber-Attack Against Ukranian Critical Insfrastructure,” [Internet]; 2016.

[7] National Vulnerability Database (NVD), National Institute of Standard and Technology (NIST), https://nvd.nist.gov/, [Online].

[8] “Russian Hackers Reach U.S Utility Control Rooms, Homeland Security Official Say,” (2018, July 23). Wall Street Journal.

[9] NERC Critical Infrastructure Protection Committee (CIPC), “Security Guideline for the Electricity Sector: Identifying Critical Cyber Assets,” June 2010.

[10] A. Hahn, A. Ashok, S. Sridhar, and M. Govindarasu, “Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid,” Smart Grid, IEEE Transactions on, vol. 4, no. 2, pp. 847855, 2013.

[11] A. Ashok, A. Hahn, and M. Govindarasu, “Cyber-physical security of Wide-Area Monitoring, Protection and Control in a smart grid environment,” Journal of Advanced Research,vol.5, pp.481-489, 2014.

[12] A. Ashok, P. Wang, M. Brown, M. Govindarasu, “Experimental Evaluation of Cyber Attacks on Automatic Generation Control using a CPS Security Testbed,” Power and Energy Society General Meeting, 2015 IEEE, july 2015, pp. 1 5.

[13] A. Srivastava, T. Morris, T. Ernster, C. Vellaithurai, S. Pan and U. Adhikari, "Modeling Cyber-Physical Vulnerability of the Smart Grid With Incomplete Information," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 235-244, March 2013.

[14] NERC, Remedial Action Development Definition Development project 2010-05.2 Special Protection System.

[15] J. McCalley et al., “System Protection Schemes: Limitations, Risks, and Management, “ PSERC, Dec. 2010.

[16] I. Kamwa, S. R. Samantaray and G. Joos, "Catastrophe Predictors From Ensemble Decision-Tree Learning of Wide-Area Severity Indices," IEEE Transactions on Smart Grid, vol. 1, no. 2, pp. 144-158, Sept. 2010.

[17] Ashok, Aditya, "Attack-resilient state estimation and testbed-based evaluation of cyber security for wide-area protection and control," (2017). *Graduate Theses and Dissertations*. 15252.

[18] V. Kumar Singh, A. Ozen and M. Govindarasu, “Stealthy cyber attacks and impact analysis on wide-area protection of smart grid,” *2016 North American Power Symposium (NAPS)*, Denver, CO, 2016, pp. 1-6.

[19] BAL-001-2 Real Power Balancing Control Performance Standard. North American Electric Reliability Corporation (NERC), February 2013.
[20] R. J. G. B. Campello, D. Moulavi, A. Zimek, and J. Sander, “Hierarchical density estimates for data clustering, visualization, and outlier detection,” ACM Trans. Knowl. Discov. Data, vol. 10, no. 1, pp. 5:1–5:51, Jul. 2015.
[21] P. Wang, M. Govindarasu, A. Ashok, S. Sridhar and D. McKinnon, "Data-Driven Anomaly Detection for Power System Generation Control," *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, New Orleans, LA, 2017, pp. 1082-1089.