WISP: Watching grid Infrastructure Stealthily through Proxies

Final Technical Report

Oct. 31st 2022

Performing Organizations:

Raytheon Technologies Research Center (RTRC)

University of Tennessee, Knoxville (UTK)

Pacific Northwest National Laboratory (PNNL)

WISP Team:

RTRC: Lingyu(Lynn) Ren, Ahmad Osman, Fu Lin, Nai-yuan Chiang, Mucun Sun, Ryan Melville, Fragkiskos Koufogiannis, Mark Moulin, Timothy Wagner

UTK: Fangxing(Fran) Li, Qiwei Zhang, Kevin Louis Tomsovic, Jinyuan Sun

PNNL: Abhishek Somani, Sohom Datta, Xueqing Sun, Milan Jain

Industry Advisor: David Bertagnolli

Reporting Period: 03/25/2019-07/31/2022

ACKNOWLEDGMENTS

This material is based upon work supported by the Department of Energy under Award Number(s) DE-OE0000899.

DISCLAIMER

Executive Summary

The complex interdependencies of cyber systems (sensors and communications), physical grids and associated electricity market operations make protecting electric power grids a significant challenge. The energy sector is constantly under new, targeted, advanced and dangerous cyber-attacks that have the potential to result in the loss of human life. These threats are further exacerbated by our need to modernize the grid. One focus of cyber security research in smart grids is the securing of the SCADA system through advanced intrusion detection systems (IDS) and bad data detection algorithms in state estimation. These methods either require full knowledge of the system topology and parameters or fail to understand the physical behaviors under attack.

WISP (Watching grid Infrastructure Stealthily through Proxies) is designed to provide additional protection to the power grid using only publicly available data. In particular, WISP exploits the spatio-temporal nature of the real time locational marginal prices (LMPs), in conjunction with other information such as bids, weather, outages and load data to analyze anomalous power pricing behaviors and then correlate those observations to localize regions of interest and identify potential cyber events. WISP is non-intrusive as the tool is deployed as a service in the Cloud or on premise and provides reliable information to system operators for enhanced situational awareness, without impeding energy delivery functions.

The WISP technology comprises three modules: the data-driven anomaly detection core, the vulnerability and risk analysis and the root cause analysis. The data-driven anomaly detection core performs the tasks of feature selection, anomaly detection and attack region

localization. The vulnerability and risk analysis module provides system level information of the vulnerable variables and times, assisting the operators in selecting monitoring and protection nodes. The root cause analysis module takes the detection results and identifies potential operational conditions that contribute to the detected anomalies.

In Phase I, we have demonstrated the feasibility and effectiveness of WISP. We developed a realistic electricity market simulator capable of generating normal and attack market data under various operational conditions. We developed a series of cyber-attack detection and analysis algorithms and evaluated them under multiple data sources. Finally, we integrated all modules into an end-to-end software, providing functions for data management, data analytics and visualization. Specifically, we have achieved: (i) real-time data acceptance from external utility interfaces with >99% acceptance rate; (ii) high performance anomaly detection algorithms with >98% detection accuracy and <0.1% false alarm rate; and (iii) ultra-low computing delay <50 milliseconds. Additionally, our team developed algorithms to identify the vulnerable variables in electricity market operations and root cause analysis functions to identify major contributors to the price spikes. These ancillary modules are necessary when deploying WISP in real world industry environment.

In Phase II, we have demonstrated the effectiveness of WISP software on realistic large-scale power systems. We performed red team testing for the Phase I WISP software and identified software vulnerabilities and implemented corresponding mitigation solutions. We adapted the electricity market simulator for the Texas synthetic 2000-bus system and generated datasets for the false data injection attacks. We created database and visualization interfaces for the Texas system and the ISO New England system. We performed software optimization in terms of operation efficiency, computing speed and detection accuracy. Finally, we tested the software on the Texas system and the ISO New England system and evaluated the detection performance. Overall, we achieved above 89% detection rate, below

3% false alarm rate and below 37 seconds of end-to-end detection delay.

In conclusion, we have accomplished the objectives for WISP research and successfully demonstrated the WISP software on large-scale systems.

Contents

§Phase I §

List of Figures

List of Tables

Chapter 1

Introduction - Phase I

The complex interdependencies of cyber systems, physical grids and associated electricity market operations make protecting electric power grids a significant challenge. Recent reports and surveys show that the energy sector is constantly under new, targeted, advanced and dangerous cyber-attacks that have the potential to result in the loss of human life. Examples include advanced cyber intrusions such as the BlackEnergy, Havex, and Sandworm malware variants that targeted critical electric power infrastructure cyber assets, including Supervisory Control and Data Acquisition (SCADA) systems. The threats against critical infrastructure from criminal groups, hackers, disgruntled employees, nation states and terrorists, whether targeted or opportunistic, are evolving and growing (see incidents reported by the Industrial Control Systems Cyber Emergency Response Team (ICS CERT)). The cyber security threat to the energy sector is not new as the U.S. Department of Energy (DOE) has led strategic road mapping activities to address cyber security threats and improve cyber resilience since 2004. The energy sector has also made significant strides in protecting the critical cyber assets at power generation facilities through the development and enforcement of standards such as Critical Infrastructure Protection (CIP) by the North American Electric Reliability Corporation (NERC).

The threat to the electric sector is exacerbated by our need to modernize the grid. As current power systems advance from a macro utility-centric model to a distributed structure, driven by the energy revolution, several new schemes such as smart metering, real-time pricing, managing demand side flexibility and distributed renewable energy resources, shall come to fruition. Such technologies will no doubt improve the operations of the grid and the efficiencies of the associated markets. On the other hand, it will also increase system exposure, providing newer entry points for hackers to disrupt grid operations. In this chapter, we review the related work in protecting electricity grids against cyber-attacks and introduce the general framework and research goals of WISP.

1.1 Related Work

One focus of cyber security research in smart grids is the securing of the SCADA system. Since the legacy system in SCADA cannot be updated or patched through traditional IT security technologies, a series of intrusion detection systems (IDS) are introduced as a countermeasure. One IDS design is statistic-based anomaly detection which identifies misbehaviors in network traffic by inspecting the headers of the packets under industry standard protocols [6]. Another category is the SCADA-specific IDS which relied heavily on domain knowledge, such as state-based IDS [7], model-based IDS [8], rule-based IDS [9] and behavior-based IDS [10]. The performance of the IDS is limited due to the following reasons: (1) the unaddressed false alarms in statistic-based IDS (2) the need for expert knowledge of the system and the tailoring of the IDS for different systems, and (3) the lack of well-considered attack models.

A few efforts have been explored to defend against false data injection attacks. In [11], efforts have been made to develop computationally efficient heuristics to detect these false data attacks against state estimation. For data injection attacks in the state estimation layer, one major approach is to harden the physical layer: either protect the basic measurements or introduce state validation through installation of advanced meters, such as PMUs. Since securing all meters is not cost-effective, efforts have been done to strategically select a subset of critical meters to keep network observability and minimize the attack effort. Algorithms are also developed to decide minimum deployment of PMUs to validate the critical states. Another technology to defend against data injection attacks is to design a historian-based detector. Instead of the estimation residue based bad data detector, a likelihood ratio test detector can be used taking advantage of the prior information to preserve and trace the likely states in the system. Data analytical approaches such as Neural Networks have been introduced to mitigate false data injection attacks [12]. The drawbacks of existing protection technologies include: (i) high cost and intrusive techniques prohibiting faster adoption of physical layer protection; (ii) potential compromise of the defenses such as GPS spoofing for PMUs; (iii) complicated algorithms that requires full knowledges of the system topology and parameters; and (iv) new attacks designed to bypass the data-driven detectors via studying the statistical behavior of the measurements.

In summary, *there are almost no efforts that focus on system-level cyber defense against multiple attack vectors, nor on using market layer observation to detect attacks at physical layer.* WISP will bridge this gap by providing a novel non-intrusive anomaly detection solution that leverages publicly available market data. WISP will analyze the advertised locational electricity price data with other information such as load patterns and system outages to detect and distinguish cyber-attacks from normal system events and also localize the region of disturbance.

1.2 WISP

The core focus of WISP is to observe publicly available prices to identify and explain the cause of anomalous pricing behaviors, whether it is due to intentional or non-intentional acts on the underlying power system and market interfaces. Phase I of WISP is executed as follows. The initial task is to delineate the attack landscape for energy markets, including intentional and non-intentional attacks and use power system and market simulation tools to generate locational marginal prices (LMPs) for various IEEE bus structures to assist the development of unique signatures corresponding to attacks and normal disturbances.

Subsequently, machine learning algorithms, using power system and market knowledge, are developed and then tested and evaluated for their ability to detect and flag anomalous events by observing variations in real time prices and other external information. The knowledge gained in the simulation environment are used to develop a real-time energy market monitoring tool that learns correlations between prices for different locations, in the presence of congestions and use it to detect and alert system operators, when anomalous deviations are observed. The end output of Phase I is an anomaly detection software that detects and alerts operators on identification of anomalous price deviations. Confidence scores and adaptive thresholds are derived to reduce the impact of false alarms.

The Phase I of the report is organized as follows. Chapter 2 presents the study of the threat and attack landscape in energy systems. Chapter 3 presents the electricity market simulator including power system operational functions, cyber-attack implementations, case study and signature derivations. Chapter 4 presents the anomaly detection algorithms and evaluation results on various datasets. Chapter 5 presents the algorithms of the vulnerability and risk analysis module. Chapter 6 presents the algorithms of the root cause analysis module. Chapter 7 presents WISP software architecture design and software development. Chapter 8 presents the commercialization plan. Chapter 9 concludes Phase I. The overview of the key objectives and achievements of WISP Phase I are presented below.

The key objective of WISP is to research and develop:

- A non-intrusive cyber-attack monitoring tool that uses readily accessible market data (LMPs) to detect and reason over anomalous pricing behaviors in order to provide insights to utilities/ISO about potential cyber events
- A library of LMP-based signatures to distinguish natural grid events from intentionally induced cyber events (e.g., corrupted grid measurements)
- Optimal detection thresholds that forces an adversary to induce minimal perturbations to the measurements or market information, thereby leading to minimal system losses

The major accomplishments of the WISP Phase I are:

- Published results in leading conferences and journals (5 papers)
- IP on "Systems and methods for anomaly detection in electric power grids using electricity market data" filed on March 2021
- Outreach via on-site and remote presentations to ISO-NE and PJM in 2020
- Completed the milestones and submitted required deliverables
- Completed DOE peer review in 2020 and live demonstration in 2021
- Developed algorithms for WISP data-driven detection core, vulnerability and risk analysis module and root cause analysis module
- Integrated data downloading, simulation, query, and storage functions with data analytics and visualization platform to a market monitoring tool capable of:
  - detecting anomalies with $> 98\%$ detection rate and $< 0.1\%$ false alarm rate
  - accepting real-time utility data with $> 99\%$ acceptance rate
  - making real-time detection with delay less than 50ms

Chapter 2

Threat and Attack Classification for Energy Markets

The increasing frequency of cyber intrusions in the modern power grid is becoming a nationwide concern and challenge. The new trend of malware attacks makes large scale cyber attacks possible in the industrial control system (ICS) environment. The adoption of advanced smart grid technologies introduces new access points for threat agents. Electricity market, as one of the core power system components, interacts with all domains ranging from entities of power generation to consumption. Electricity price is a synthesized reflection of the physical system, market strategy, customer behavior and weather conditions, etc. Impactful attacks, either target on the market or physical system, will compromise the input of the market machine and lead to unpredictable and undesirable shifting and spikes in electricity price. This chapter depicts the details of these threats and attacks classified by their technology areas and attack interfaces.

2.1 Introduction

Electric distribution infrastructure, powering communication, transportation, health, water treatment, etc., is at the heart of all critical public serving facilities. It is an urgent task [13–15] to enhance grid resilience against the ongoing and emerging advanced cyber intrusions at the energy sector as shown in Figure 2.1. Malware, malicious software, is an effective and most frequently reported manner to compromise industrial control systems (ICS) as proven by cyber incidents all over the world. Successful attacks involve combined exploits of multiple vulnerabilities spread across all information technology (IT) and operational technology (OT) components. Among these exploits, phishing and watering hole are frequently used in the first stage of cyber intrusion and malware is often deployed to take control of the victim system [13]. Denial of service (DOS) is another popular and continuously reported attack revealed by the data from the Industrial Control Systems Cyber Emergency Response Team (ICS CERT).

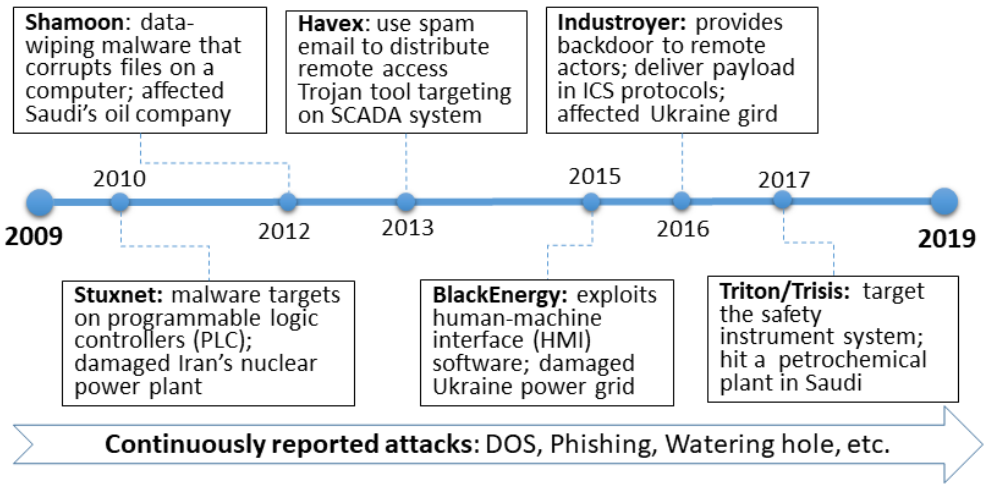


Figure 2.1: Recent Trend of Cyber Attacks in Energy Sector

In the battle against cyber threats, one major accomplishment is the establishment and promotion of the North American Energy Reliability Corporation (NERC) Critical Infrastructure Protection (CIP) guidelines, which are mandatory for bulk power systems. However, as of year 2015, only an estimate of 10%-20% grid assets are covered by CIP [16] since most distribution companies are under state or local regulations. Meanwhile, the Department of Energy (DOE)'s grid modernization effort has further advanced the automation and digitalization of the bulk power grid and elevated the utilization of latest distribution grid technologies, such as distributed energy resources (DER), advanced metering infrastructure (AMI), and demand response (DR). Though this movement enables great efficiency and flexibility, it also opens additional access points to power grids accompanied by the new vulnerabilities and attack vectors. This chapter will discuss the details of these vulnerabilities in section 2.3 in category of their technology areas.

The adversary activities in power systems also affect the energy market since the market machine (a set of algorithms that generates electricity prices and generation dispatch decisions) is highly coupled with the underneath operational and physical systems. Information extracted from the electricity market data can be used to indicate the operational conditions of the power system and detect malicious cyber incidents. Currently, there is a lack of effort in evaluating the market impact of the cyber attacks and using market data as an indicator.

To fill the gap, we developed a market monitoring tool, WISP, to analyze anomalous power pricing behaviors and then correlate those observations to localize regions of interest and identify potential cyber events. In WISP framework shown in Figure 2.2, we consider the baseline system as a three-layered model including power system layer, energy management layer and market management layer. Under this model, three classes of cyber attacks are considered: *(1) the false data injection attacks* that compromise data integrity of the mea-

surements, *(2) the physical response attacks* that intrude the grid edge devices and power users, *(3) the market interface attacks* that exploit the bids and offers.

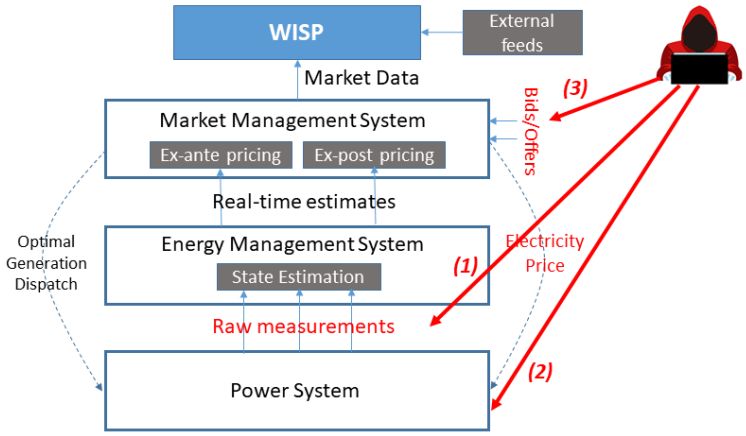


Figure 2.2: Overview of the WISP framework

The remainder of this chapter is organized as follows. Section 2.2 presents baseline energy market model and the supporting technologies. Section 2.3 presents common vulnerabilities and threats in energy systems and the attack scenarios that impact the energy market. Section 12.3 concludes this chapter.

2.2 System Model

The operation of electricity market, specifically the wholesale market, depends on the cooperation of five major participants: generation, transmission, distribution, customer and service provider [17]. The operation center (OC), i.e. Regional Transmission Organization (RTO) or Independent System Operator (ISO), maintains the power balance using energy management modules, e.g. security constrained economic dispatch (SCED), unit commitment and contingency analysis, and market management modules, e.g. day-ahead market and real-time market. The bulk power plants (generation) join the market as suppliers. They submit their offers (quantity and price) to OC and receive dispatch decisions from OC. The transmission system moves power from generation to distribution usually in high voltage and long distance. Transmission system is monitored, controlled and protected via sensors, relays, and circuit breakers, leveraging technologies such as Supervisory Control and Data Acquisition (SCADA) and Phasor Measurement Unit (PMU). The measurements from transmission lines and substations are sent to the OC in real-time to support dynamic dispatching and pricing through remote connections. The distribution system delivers power to the customers. Small commercial and residential customers only join the retail market while the large industrial customers can directly participate in the wholesale market. The

service providers, utilities or third party providers, report the aggregated demand to the OC and bid for the suppliers' offers. They provide wide range of services such as billing, accounting, communication and control of energy use. The concept and interactions of the market participants are illustrated in Figure 2.3.

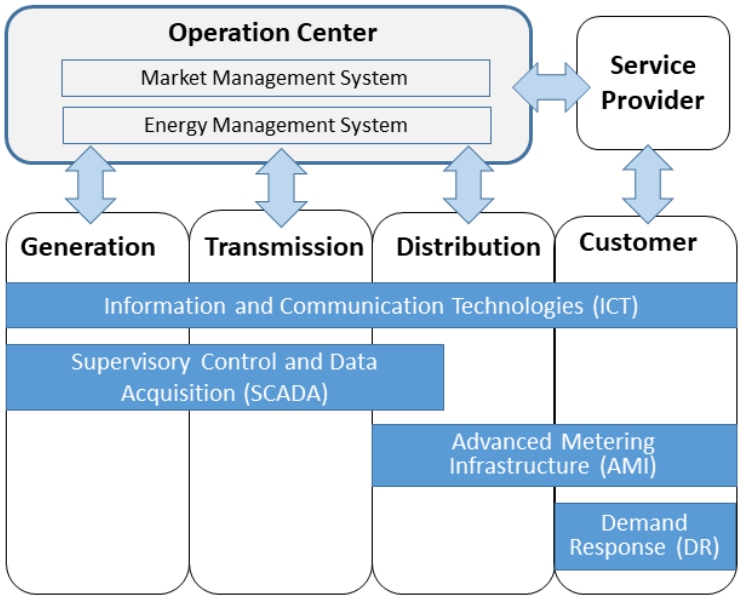


Figure 2.3: Technologies Supporting Power Grid and Electricity Market

In the system model, we consider several critical and vulnerable technology areas that support smart grid and electricity market. In particular, the information and communication technologies (ICT) refer to all networking and computing devices, firmware, software and protocols that enable two-way information transmission, logging and storage. ICT is the fundamental infrastructure that serves all smart grid applications, of which we focus on SCADA, AMI and DR since they directly contribute to electricity pricing.

SCADA system mainly consists of three elements: the remote terminal unit (RTU), the master terminal unit (MTU) and the human-machine interface (HMI). RTUs are widely installed in substations as data acquisition equipment for field sensors, and executors of commands from MTU. MTUs, located in master stations, gather data from RTUs and report to the operators through a graphic interface, HMI. MTUs are responsible for data logging, alarming, trending and reporting, and also provide control interface to the field devices, such as circuit breakers and switches. Two major protocols used for the remote connection of RTUs and MTUs in power grids are distributed network protocol V3 (DNP3) and IEC 61850. Traditional SCADA system does not measure synchronous phasor signals, thus the state estimation is needed to calculate power system states for power planning, contingency analysis, economic dispatch and marketing.

While SCADA provides monitor and control for industrial systems (power plants, transmission and distribution systems), AMI is designed for commercial and residential environments and is responsible for two-way communication between customers and service providers. AMI consists of (1) smart meters which collect hourly or more frequent user power consumption data, (2) headend server which aggregates user data and performs data analytics, and (3) communication network. AMI enables more accurate load measurement and load forecasting for electricity market.

Electricity market leverages price incentives to maintain power balance. One recent marketing technology is the demand response (or demand-side management) program which changes consumption patterns in response to price of electricity. DR can be (1) incentive-based scheme where the service providers schedule, reduce or disconnect the loads at high price, or (2) price-based scheme where the service providers distribute the varying prices and the customers individually adjust their power usages [18].

2.3 Adversary model

The U.S. power grid is a complex cyber-physical system incorporating vast volume of distributed devices, which by nature results in a large attack surface. Malicious attacker, targeting on local outages, equipment damages, grid instabilities or individual financial gains, can compromise the power devices, communication and control facilities or market interfaces (web service to the customers and suppliers), as shown in Figure 2.2. For such complicated system, single exploit can hardly succeed in conducting effective cyber intrusions and a series of steps must be taken. These steps, also called ICS Cyber Kill Chain [19], contains two stages: stage one is to conduct reconnaissance, exploit attack vectors, deliver payloads and escalate privileges; stage two is to explore the ICS environment and create ICS effects. The starting point of all adversary activities is to understand the target environment and its vulnerabilities. We have introduced briefly the system model in the previous section. In this section, we will first review the common vulnerabilities in the focused technology areas, followed by a detailed delineation of the attack scenarios and their impacts.

2.3.1 Vulnerabilities and threats in smart grid

Vulnerabilities introduced from the supporting technology areas are elaborated in the Table 2.1, referring to [20–24].

Threat agents can exploit these vulnerabilities to conduct sophisticated attacks with the objectives of loss, denial or manipulation of view, control or safety of the target system. Popular exploits, to name a few, include malware (viruses, worms, Trojan horses, etc.) attacks, denial of service (DOS) attacks, man-in-the-middle (MITM) attacks, reply attacks, jamming channels, popping the HMI, data integrity and privacy violations. In particular, in the MITM attack, the threat agent inserts a relay device (hardware or software) between two

Table 2.1: System Vulnerabilities categorized in technology areas

	Communication and Networking	Software and Firmware
SCADA	lack of bounds checking, buffer overflow possible, weak authentication and no encryption in network protocols, lack of network segmentation, weak protection of user credentials, access to ports not restricted, potential remote access through virtual private network (VPN)	unpatched operation system, unauthorized directory traversal allowed, services running with unnecessary privileges, use of potentially dangerous functions in the code
AMI	inadequate protection for Internet access, easy physical access, weak or no cryptography on internal bus, wide use of symmetric key, insecure key storage, insufficient integrity protection, inadequate network segmentation, commands replayable, lack integrity protection in cellular network, weak or no authentication to home area network (HAN), poor time synchronization check	weak or no authentication to install firmware/software, no detection of unauthorized installation, weak authentication and security configuration to database software, weak credentials in meter settings, shared passwords and credentials
DR	easy physical access, wide use of the same cryptographic key, lack of data source validation, unnecessary open ports, lack of network monitoring	inadequate access control to configuration files, out-of-date patches and anti-virus signatures,

legitimate parties and interferes with the traffic between them. Authors in [25] investigated the vulnerabilities in DNP3 protocol and performed experimental MITM attacks to alter the DNP3 payload encapsulated in a TCP/IP packet between the RTU and MTU. Based on the likelihood and severity analysis in [26], malware attacks and DOS attacks are the two high-impact-high-probability cyber incidents in smart grids, consistent with the findings in ICS CERT reports [27].

2.3.2 Attack Scenarios

For this project, we consider three attack scenarios categorized by their attack targets as shown in Figure 2.2. For each scenario, we explain below the attack objective, attack design and attack impact, supported by a review of related work.

False Data Injection Attacks (FDIA)

Attack Objective: The objectives of the FDIA attack include (1) market manipulating and (2) operational disturbances. The attacker could create false load estimation or transmission congestion limits to mislead the real-time electricity pricing algorithms in producing biased locational marginal prices (LMP). The attacker then takes advantage of the biased LMP to gain monetary profits using bids and offers via the market interface. The FDIAs, aiming at disrupting the system operation, is often state funded or terrorist activities, which intend to break down the critical infrastructure of the target region. The adversary could inject bad measurement to deceive the SCED to make insecure dispatch decisions where transmission lines are overloaded such that relays are triggered and local power outages or equipment damage occur.

Attack Design: The theory behind FDIA is based on the power system state estimation which estimates state variables through real-time measurements provided by SCADA and

grid network models. The mathematic formulation of FDIA is shown as Eq. 2.1.

$$\begin{aligned}\hat{x}_{bad} &= (H^T W H)^{-1} H^T W \cdot z_{bad} \\ &= (H^T W H)^{-1} H^T W \cdot (z + a) \\ &= (H^T W H)^{-1} H^T W \cdot z + (H^T W H)^{-1} H^T W \cdot a \\ &= \hat{x} + c\end{aligned} \tag{2.1}$$

Eq. 2.1 maps the injected false measurements (z_{bad}), e.g. voltage magnitude and power generation, to the state variables ($\hat{x}_{bad}$), e.g. voltage angle and power flow. The grid topology and parameters are incorporated in the Jacobian matrix H and the measurement errors are depicted in W. The attack design is to craft an attack vector a so that the state variables will be shifted by a desired value, c.

The FDIA requires two assumptions: (1) attacker has knowledge of the target power system and (2) attacker can compromise a large amount of measurements. For the first assumption, the attacker could use estimated topology results or partial information to derive a close estimation of H. One way is to use public available LMP to identify grid topology [28]. For the second assumption, it is hard to compromise a large number of measurement devices since they are geographically scattered. There are efforts focusing on optimal attack design to reduce the attack effort by carefully select vector a. Another group of attack design targets on the topology meters, such as the circuit breaker status sensor. The ON/OFF status of the transmission lines will restructure the H matrix. In the sophisticated attacks, tampering with topology meters can also influence power system operation.

Related Work: A number of papers along the lines of attacking state estimation exist. In [29], the scheme of false data injection attack against power state estimation was first introduced, under the assumption that the attacker can access the current power system configuration information and manipulate the measurements of meters at physically protected locations such as substations. By leveraging the knowledge of the power network topology, it was shown that one could construct false data that could bypass the bad data detection in today's state estimation system. Following this seminal paper, many efforts were proposed to quantify the efforts required to implement such a class of attack with least effort, that is derive the attack strategies in terms of the type of the meters attacked, the minimum number of meters required and also the minimal knowledge of the power network topology. For instance, the load redistribution attacks in [30] defines a special type of data injection attacks in which the load bus injection measurements and line power flow measurements are attacked. These attacks assume limited access to measurement meters and the effect is to increase load at some buses and reduce loads at other buses while maintaining the load unchanged. As a special case of false data injection attacks, the load redistribution (LR) attacks can mislead the state estimation process without being detected by any of the existing techniques for bad data detection.

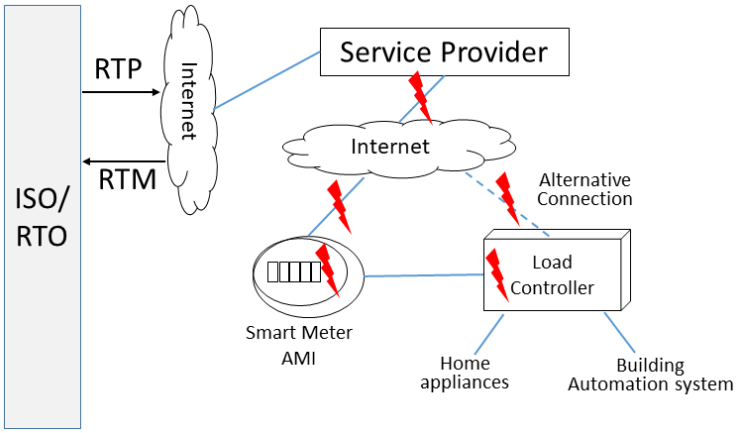


Figure 2.4: Demand response attacks targeting the real-time pricing (RTP) and real-time measurement (RTM) feedback loop

Very few efforts have devoted their attention to study the effect of cyber-attack on financial markets. Xie et.al [31] first presented the impact of false data injection attack on electricity markets. Leveraging the false data injection in state estimation, the electricity market can be manipulated to fulfil some malicious financial tasks. [32] proposed using virtual bidding at selected buses to achieve continuous financial arbitrage. The LMP of these chosen buses are influenced by the strategically crafted attack vector in state estimation in the way to gain financial benefits during virtual bidding. [33] presented a sophisticated attack aiming at making profit for the generator at a specific bus by fabricating a biased transmission congestion pattern and thus manipulating the price at a certain bus. [34] further proved that by fabricating a fake transmission congestion pattern, false data injection attacks can manipulate the real-time price at an arbitrary target bus.

Physical Response Attacks

Attack Objective: In physical response attacks, attackers leverage the vulnerabilities and easy access of grid edge devices to create certain power usage patterns, with the objective to destabilize the system operation or marketing. The maintenance of power balance relies on accurate load forecasting which is used for real-time generation dispatch. The physical response attacks compromise the load control system and deviate load from its historical behavior. Such unpredictable changes, when designed against system regulations, could drive the states to exceed stability boundaries and disrupt system operation.

Attack Design: There are multiple pathways to conduct physical response attacks and we focus on the malicious demand response program as shown in Figure 2.4. In the incentive based DR scheme, such as direct load control, the attacker could compromise

the service provider's server or the network connection with the customer side devices, e.g. smart meter or load controllers, to send malicious control signals. In the price based DR scheme, the attacker could inject falsified price data or install malware in load controllers. The assumption of this attack is that the adversary can compromise a large amount of load so that the attack is sufficiently severe to disturb system operation. Examples include inserting a delayed or scaled price signal to create sudden demand peak, planting malware in load controller to shift load to a certain time interval, and sending disconnect signal from management server to massive loads to create abrupt load-generation gap. These attacks are in the commercial or residential environment and are relatively easier to achieve compared with the well-guarded industrial environment.

Related Work: The security of demand response algorithms with real-time electricity pricing has been the subject of recent research efforts. Most of those works focus on *integrity attacks* where the adversary manipulates the real-time prices through for example scaling or delay of prices in an attempt to destabilize the market price or to shift the clearance price to regions that lead to a wider gap between supply and demand. Roozbehani *et al.* in [35] model the real time pricing process as a control model and analyze its stability conditions. Giraldo *et al.* in [36] study attacks on real time price under robust control theory framework to design a sequence of add-on items that causes the maximum generation-demand gap in the grid. Tan *et al.* in [37] study the stability of prices under real time pricing attack with price scaling and delay attacks. Li *et al.* in [38] show that even with a random backoff scheme, in which each power consumer chooses a random time to change its power response, the attacker can cause significant change in the clearance price hence maximizing the gap between the demand and supply. Yang *et al.* in [39] propose a new distributed real-time pricing algorithm and analyze the real-time prices under compromised power generation and baseline prices. Barreto *et al.* in [40] provide mitigation measures to data integrity attacks that compare consumption with historical user consumption. Under integrity attacks, it is possible to manipulate prices to gain *economic* advantage. Barreto *et al.* in [41] study dynamic price control schemes to guide the dynamic real-time price toward the attackers best profits. Liu *et al.* in [42] allow attacker to change the guideline price that mislead other customers so as to benefit the attacker under a demand-response mode. Similarly, Wei *et al.* in [43] apply a similar technique tailored to buildings as opposed to houses in [42].

While integrity attacks can potentially impact the stability of the grid, their effect is indirect and is largely influenced by the robustness of the ISO pricing measures. *Load altering attacks*, on the other hand, aim to directly destabilize the grid with circuit overflow or other adverse effects through compromising certain unsecured controllable loads. Mohsenian-Rad *et al.* in [44] provide direct and indirect attacks that leverage grid information to decide cost-effective portion of protected load. Amini *et al.* in [45] leverage frequency information as a feedback to calculate the load increase that can cause grid instability. Ryotov *et al.* in [46] mitigate the risks of load altering attacks by providing a policy-based anomaly detection algorithm that relies on the smart meter data to show alerts whenever abnormal or unsafe

operation conditions, low power quality conditions, or violation in customer policies are detected.

Market Interface Attacks

Attack Objective: Market interface refers to the web services provided by the operation centers (OC) to publish prices to and receive bids/offers from the market participants. The objective of the market interface attack is to manipulate the electricity market to gain financial profits or generate social chaos with sharp price changes and even widespread outages.

Attack Design: With the trend of open market, small power plants are allowed to participate in wholesale electricity market. In two-settlement markets, financial entities can also join the market as virtual traders who buy electricity from day-ahead market and sell the same amount in real-time markets, or vice versa. Attackers could first register as a legitimate market player and then strategically trade in the virtual market to arbitrage on the price differences. Market interface attack can be combined with the FDIA as introduced in [32]. Another example of market interface attack is the malicious modification of the bids [47] in the day ahead market in order to create transmission congestions in real time market and gain desired high LMPs for certain suppliers. Electricity market manipulation is a big concern and could diminish social welfare and affect people's daily life as proven by the California energy crisis in 2000 [48].

It should be noted that the market monitors closely observe all market related activities. For instance, the true generation cost, based on prevailing fuel cost, is well-known to market monitors, and hence, they closely monitor participants' bids to make sure those are within some tolerance bands. Hence, the market manipulation opportunities based on just strategic bidding are fairly limited under stringent market regulations. To make the bidding modification effective, the attacker could first identify the current marginal units and modify just the bids of these units. The major challenge to the market monitors comes with increasing wind and solar generation, where there is no fuel cost, and hence, the conventional methods to assess the exercise of market power don't work. One way that these resource owners can influence market outcomes is through physical withholding, i.e., under/over forecasting day-ahead relative to actual generation potential. Such a strategy can be leveraged by the threat agents to reshape market prices toward a desired pattern.

Related Work: There are very few publications related to this topic which have been mentioned above.

2.3.3 Attack Impact

Cyber attacks on power grids induce both market impact and operational impact. Most of the papers focus on the operational impact of cyber events, such as power outages, volt-

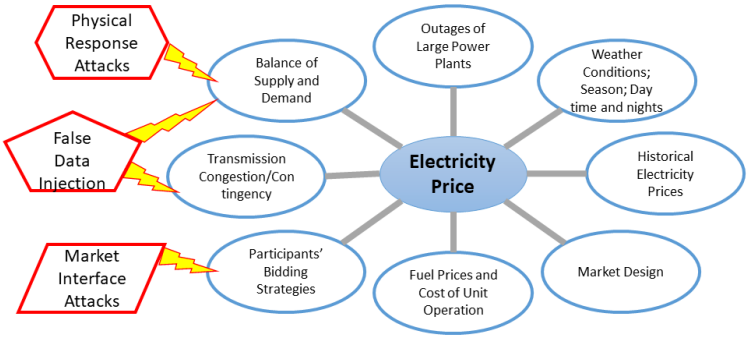


Figure 2.5: Market Impact of Cyber Attacks

age/frequency violations, line overload, and system instability (cascading failure). Specifically, for the FDIA, false data will lead to a false SCED solution that may harm power system operation in two steps. First, it may lead the system into a non-optimal generation dispatch; which at worst case can lead to load shedding. Second, it may lead the system into an in-secure operating state, i.e., power flows on some transmission lines may actually exceed their capacities. Without immediate corrective actions, the outage of these overloaded lines will cause wider load shedding in a delayed time. The physical response attacks, on the other hand, directly interfere with the power consumption devices on the grid edge to create large demand-generation gap and push the system to its stability boundaries.

The market impact of the cyber attacks is often underestimated and not well discussed in the literature. Figure 2.5 lists all the elements that contribute to the electricity price and maps them to the three classes of attacks. Whether the adversary intends to affect the market or not, the electricity price will change patterns since it is sensitive to and easily affected by any power grid or business disturbances. The impact of attacks depends on "when and how" the attack is conducted, i.e. is the grid weakly operated now? are the affected devices, single or aggregated, significant enough? These questions can be answered by the attack scenario design mentioned above. A successful attack should be stealthy, effective, and persistent, which is often achieved by national advanced persistent threat (APT) teams.

2.4 Conclusion

The power grid is moving towards an intelligent, automated and sustainable smart grid scheme, integrated with advanced applications such as SCADA for grid monitoring and control, AMI for real-time load monitoring and DR for market balancing. These applications oversee large scales of sensors and actuators through remote connections which inevitably opens entry points for intruders. Threat agents can target on the end users, the grid facilities or the market interface. Based on this, we categorize the potential attacks into three classes: the false data injection attacks, the physical response attacks and the market interface at-

tacks. To monitor, detect, and reason these attacks, we developed a non-intrusive market monitoring tool: WISP-Watching Grid Infrastructure Stealthily through Proxies. WISP will have a broad impact in the electricity industry in solving the long-term challenges of system-level cyber defense.

Chapter 3

Dataset Generation and Signature Derivation

Generating realistic datasets and deriving event signatures are critical to the success of data-driven algorithm development and evaluation. This chapter delineates the theory and procedure of establishing a practical electricity market simulator, consisting of major functions in real-world systems, such as state estimation, economic dispatch, reserve market and demand response. Additionally, we implemented three types of cyber attacks, i.e., load redistribution attack (LRA), price responsive attack (PRA) and false data injection attack (FDIA), and analyzed their long-term and short-term impacts to the electricity market. We compared the system response under cyber attacks and under operational events to identify the unique cyber attack indicators. Test results and observations are detailed in this chapter.

3.1 Introduction

In Chapter 2, we researched the threats and vulnerabilities of smart grids and investigated the three groups of cyber attacks, i.e. the physical response attacks, the false data injection attacks and the market interface attacks. Task 3 is launched based on the findings of Task 2 with the clear goal of providing practical and representative datasets and signatures for Task 4. To achieve this goal, we addressed three major challenges: (1) the lack of time series full-cycle electricity market simulators; (2) the lack of cyber attack impact analysis on long term simulation; and (3) the lack of comparison between normal operational events and cyber attacks.

The evolution of modern power systems is largely reshaping the research and field practice of the power industry. One example is the electricity restructuring which introduces the competitive market mechanism to encourage energy efficiency and innovation. Nevertheless, this transformation led to the California Energy Crisis in 2000, which necessitated the market regulations and pre-testing of market decision-making algorithms through accurate simulation models. Great effort has been invested to develop models of the restructured electricity market. A comprehensive list of power system analysis software is provided in [49] and the open source software is especially given in [50] for research, teaching and training purposes.

Table 3.1: Comparison of Open Source Electricity Market Simulator

	State Estimation	OPF	IED	Bidding Strategy	Network Model Modification	Optimization Constraints Modification	IEEE Test Case	Large-scale Test Case
AMES [51]	-	-	✓	✓	-	-	✓	-
MASCEM [52]	-	-	✓	✓	-	-	-	-
MATPOWER [53]	✓	✓	✓	-	✓	✓	✓	-
PSAT [54]	-	✓	-	-	✓	-	✓	-

In the interest of WISP, we compared a group of actively maintained open source software in their capability of supporting both standard and customer defined functions, shown in Table 3.1.

Both Agent-based Modeling of Electricity Systems (AMES) and Multi-agent Simulator of Competitive Electricity Markets (MASCEM) are agent based simulators which model all market participants as "agents" and market activities as interactions between agents. They focus more on the bidding and pricing strategies while simplifying the physical system models. Thus, the agent-based simulators cannot support the implementation of cyber attacks which requires full or partial knowledge of the system topology and critical sensor data. On the other hand, both MATPOWER and Power system analysis toolbox (PSAT) are Matlab based software supporting detailed physical models. MATPOWER is more powerful in steady-state and market-related functions, while PSAT is featured by its transient-state simulation. Both provide modularized and callable functions. Due to these merits, we chose MATPOWER as our base library and built our simulator upon it. Note, there is currently no simulator that offers cyber attack modules.

To generate time series operational data, the first step is to integrate the standalone functions into a full-cycle pipeline. In this Chapter, we built multiple interfaces and wrappers for the major elements in market operation, including state estimation, economic dispatch, demand response and optimal power flow. We used an AC/DC hybrid model to minimize the inaccuracy caused by system loses. With the baseline simulator, we then implemented three cyber attack models, i.e. the load redistribution attack (LRA), the price responsive attack (PRA), and the false data injection attack (FDIA). These attacks are profit motivated but could cause system instability and even blackouts. Their impacts are studied through short-term and long-term simulation using different parameters. Cyber attacks are easily mixed with the power system outage events considering both create dramatic changes in the system response. To understand their differences, we extended our simulator with outage management functions and reserve market interactions. This effort is to make sure the system runs continuously without being forced to downgrade its power supply services. This is also aligned with the industry standard practices. Using the simulator, we were able to create datasets with/without cyber attacks, with/without outages and compare these scenarios to derive signatures of malicious activities in the grid. The numerical results and analysis are elaborated in this chapter.

The remainder of this chapter is organized as follows. Section 12.1 presents the electricity market simulator framework and its major components. Section 3.3 presents the implemen-

tation of generator/line outages and reserve market. Section 12.2 presents the cyber attack algorithm and simulation models. Section 3.5 illustrates the experiments on IEEE test cases and the result analysis. Section 12.3 concludes this chapter.

3.2 Electricity Market Simulator

Among all the open source power grid simulation tools, MATPOWER prevails in the research of economic dispatch, optimal power flow (OPF) and unit commitment. It provides a stable version of major power system functions and supports interface modifications and customized applications. However, MATPOWER's function modules are designed independently for certain power flow snapshots. There is no capability to integrate these modules and generate time series power system states and electricity market data. As the first step of dataset generation, we built an electricity market simulator which takes load profile data and generates serial power measurements and marginal prices.

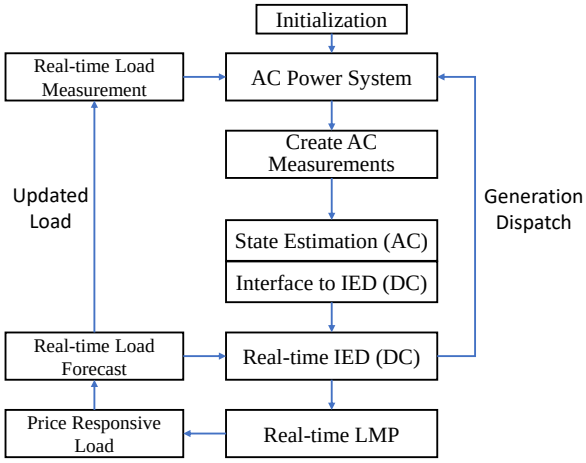


Figure 3.1: The baseline electricity market simulator.

The flow chart of different modules in this simulator is presented in Figure 3.1. For the reason of clarification, we did not include the logical modules in the figure, e.g., the module used to start or terminate the simulator according to the given simulation duration. Following the scheme of PJM and ISO-NE, we adopted a hybrid structure, where both AC and DC formulation are used. This flow chart presents the basic elements in the simulator, such as getting different measurements from available sensors and meters, and subsequently using them to estimate state variables which represent the operating condition of power systems. Based on the current status, an incremental economic dispatch (IED) model is solved with the next five-minute load forecast, to provide the dispatch plan and locational marginal prices (LMPs) for the next operation interval. The details of each module in Figure

3.1 are further explained in the following subsections.

3.2.1 Initialization

The *initialization* block in Figure 3.1 sets up the necessary environments for the simulation, such as the parameters to control the simulation length. The initial state variables are determined by solving a DC-OPF model with day-ahead hourly load forecast at the starting time of the simulation.

This module is also responsible in normalizing the input load forecast data for different test cases. For example, PJM publishes the hourly load forecast and five-minute load forecast for 25 transmission zones. These 25 sets of data are used in the simulator for the IEEE-39 system, after scaling and matching the load data to the original load bus parameters carefully. Specifically, we firstly rank the given loads in IEEE-39 system and the PJM zonal daily total load separately. We then map the PJM zonal load to the load buses according to their rankings. If a target network has more than 25 load buses, for example, with 35 load buses, we will duplicate the last ten zones in PJM. We used constant power factor to get the reactive power. By scaling the PJM data, we guarantee the power flow stays within meaningful values and not deviate much from its original solution.

3.2.2 AC Power System

The physical power system is represented as an AC power flow module in the simulator. This module takes the load measurement and generation dispatch data as input and calculates power flow (PF) for each time step. These PF results serve as the basis for incremental dispatch. The AC power system block illustrates how the power grid responds to the IED results. Specifically, we consider two factors: (1) the LMP results will be used to modify the price responsive load and thus impact the load measurement data; (2) the generation dispatch results will be used to adjust the generator power output and reallocate the system resources.

3.2.3 AC Measurements

The *Create AC Measurement* block in Figure 3.1 presents the step to simulate the measurements used by the control center. In practice, these measurements are obtained from remote terminal units (RTUs), or other sensors such as Phasor Measurement Units (PMUs). These data are considered proprietary and confidential to the ISO/RTO entities, which requires us to use public available test cases and create artificial measurements. In the simulator, we take the AC power flow solution from the *AC Power System* block, and add small random perturbations to represent the measurement noises from the sensors at each five-minute time step.

The measurements include the real and reactive power flow rate, real and reactive power

generation, voltage angle and voltage magnitude. Note that in practice, not all the measurements are available, e.g., missing of meter on a particular transmission line. This issue will be handled by the next module.

3.2.4 AC State Estimation

The state estimator is a standard power system operation tool used to provide a complete and reliable model of actual operating conditions. It uses actual operating conditions based upon available observations, e.g., from metered measurements, along with the network topology and parameters to calculate the remaining variables that are not metered [55, 56].

This module is usually implemented in a control center equipped with a SCADA system. Since inaccurate data measurements can lead to significant errors in state estimation, SCADA uses bad data detection (BDD) to test if the measurements are reliable. BDD can be implemented in a variety of algorithms, and finding a reliable and robust algorithm is a hot research topic. Since WISP focuses on the cyber attacks in power systems, not the BDD and state estimator, we adopted the existing state estimator from MATPOWER and implemented a popular BDD strategy [57, 58].

The states of an AC power system are usually defined as the voltage magnitudes and phase angles at all the buses. However, the measured data typically incorporate the active and reactive power flow at transmission lines, active and reactive power injections and some voltage magnitudes at certain buses. The relation of state variables and measurements follows this mathematical representation:

$$\vec{z} = h(\vec{x}) + \vec{e} \tag{3.1}$$

where
$\vec{z}$: measurements;
$\vec{x}$: state variables;
$\vec{e}$: measurement errors;
$h(\vec{x})$: nonlinear equations used to establishes the relationship between $\vec{x}$ and $\vec{z}$ in the AC model.

The state estimation problem is to find the best estimate $\hat{\vec{x}}$ of state variables $\vec{x}$, given all the observables $\vec{z}$. This can be modeled as an optimization problem

$$\min_{\vec{x}} \quad (\vec{z} - h(\vec{x}))^T \mathbf{W} (\vec{z} - h(\vec{x})) \tag{3.2}$$

where $\mathbf{W}$ is a weight matrix used to control the accuracy preference of the individual measurements. Usually the measurement errors $\vec{e}$ are unknown, but their distribution can be obtained, e.g., from the user manual of the measurement equipment. Therefore, one popular choice of the weight matrix $\vec{W}$ is to set $\mathbf{W} := \mathbf{Cov}^{-1}$, where matrix $\mathbf{Cov}$ is the co-variance

matrix of the measurement errors $\vec{e}$. Unlike the DC state estimation, equation (3.2) is classified as a nonlinear least-square optimization problem, and it does not have a closed-form solution. In practice, the AC state estimation problem is solved by Newton-type iterative methods, which only guarantee a local minimum.

The meter measurements corresponding to the best estimate is denoted as $\vec{\hat{z}} = h(\vec{\hat{x}})$. Hence we can define the residual $\vec{r}$ as

$$\vec{r} = \vec{z} - h(\vec{\hat{x}}) = \vec{z} - \vec{\hat{z}} \tag{3.3}$$

The value of this residual is often used as test criteria in BDD, which tries to identify faulty measurements caused by malicious attacks or equipment failures. The idea is that legitimate measurement residual $\vec{r}$ can bypass the residual test

$$\|\vec{r}\|_2 \leq \tau \tag{3.4}$$

where τ is a given threshold. If this condition holds, the measurements are used and the corresponding state estimation is trusted; otherwise we assume that at least one bad measurement exists and measurements at this time step will be discarded while that of previous step will be reused.

3.2.5 Interface to DC

DC power flow model is widely used by industry practitioners in their daily work [59]. The reason is that, for large power systems, solving nonlinear AC economic dispatch problem, or AC-OPF, is computationally expensive and only local optimal solution can be found by Newton-type iterative methods. On the other hand, the DC model is more attractive in practice. It linearizes the nonlinear non-convex AC power flow equations, with mild assumptions such as voltage magnitude are constant and the phase angle differences are small. As a result, the outcome DC formulation only contains real power, and the transmission loss can be ignored. The use of such a model leads to significantly simplified linear expressions, and state variables can be reduced to only the phase angles. Computationally speaking, the DC formulation can be solved and optimized much more efficiently, and a unique solution is guaranteed. This advantage is significant for the contingency analysis, when an enormous number of power flow equations need to be solved. Last but not the least, the linearized model with unique solution fits the economic theory much better than the AC model. The definition of LMP is based on the shadow price of DC-OPF, which is the kernel of the real-time energy market.

Therefore, we use DC formulation to plan the power dispatch for each time step. In order to use the AC state estimation for DC dispatch, we need a module to transfer the AC power flows into the DC ones. Compared to the AC formulation, one big difference is that the transmission loss is absent from the DC formulation. Hence the total real power

generation in AC formulation is larger than that in DC formulation in each time step. When we solve an AC power flow, this loss is reflected in the power generation at the reference bus (swing bus), i.e., the reference bus will generate additional power to cover the loss while keeping the other power generation fixed to their own dispatch plan. In order to use the AC state estimation results in DC model, one straightforward way is to shift all the transmission losses to the reference bus as an extra load. However, this approach will introduce a big bias to the reference bus power output, especially for large networks where the accumulated loss is huge. Therefore, we adopt the idea of *Loss Factor* and *Delivery Factor* from [60], which distributes the total transmission loss to all the load buses, proportionally to their original load values. As a result, the bias can be flattened in the full DC network.

3.2.6 DC Dispatch

As mentioned in the previous subsection, the DC formulation is less accurate, but much simpler to derive and much computationally easier to solve than the AC formulation. Therefore it is widely adopted in industry, such as ISO New England and PJM [61].

The generic lossless DC-OPF model can be modeled as an optimization problem, which minimizes the total real power generation cost subject to DC power flow balance and transmission flow limits. There are different mathematical formulations for this optimization problem [55, 60–62] based on different usages. However, they share the same definition of LMP, i.e., LMP is composed of shadow prices from the optimization problem. In our simulator, we used the PJM incremental linear programming formulation [55], with the popular concept of generation shift factor [60], for the dispatch problem and computed LMP from it. This *incremental economic dispatch* problem is presented as follows:

$$\min_{\Delta G} \quad \sum_{i}^{N} c_i(\Delta G_i) \tag{3.5a}$$

$$s.t \quad \sum_{i}^{N} \Delta G_i = \sum_{i}^{N} \Delta D_i \tag{3.5b}$$

$$\Delta F_j^{min} \leq \sum_{i}^{N} GSF_{j,i} \times (\Delta G_i - \Delta D_i) \leq \Delta F_j^{max},$$

$$\forall j \in \mathbb{L} \tag{3.5c}$$

$$\Delta G_i^{min} \leq \Delta G_i \leq \Delta G_i^{max}, \quad \forall i \in \mathbb{B} \tag{3.5d}$$

where

$\mathbb{B}$: the set of buses;

$\mathbb{L}$: the set of lines;

N: total number of buses;

M: total number of transmission lines;

$c_i(\Delta G_i)$: incremental generation cost at bus i;

GSF: generation shift factor matrix; it is an $M \times N$ matrix where the $\{j,i\}^{th}$ element presents the generation shift ratio to line j from bus i;
ΔG_i: incremental real power generation at bus i;
ΔD_i: incremental load at bus i;
ΔF_j^{max}: maximum incremental real power flow at line j;
ΔF_j^{min}: minimum incremental real power flow at line j;
ΔG_i^{max}: maximum incremental real power generation at bus i;
ΔG_i^{min}: minimum incremental real power generation at bus i.
λ: dual variable for constraints (3.5b).
$\vec{\mu}^-, \vec{\mu}^+$: dual variables for constraints (12.2c).

Based on the state estimation results, IED (3.5) only focuses on the difference between the current state and the next five minute status. The incremental real power generation $\Delta\vec{G}$ is the only optimization variable in (3.5), and therefore this linear optimization problem can be expected to have a quick and robust solution. The incremental load $\Delta\vec{D}$ is a constant parameter computed as the difference between the current load and next five-minute load forecast. The boundary constraints $\Delta\vec{G}^{min}$, $\Delta\vec{G}^{max}$, $\Delta\vec{F}^{min}$ and $\Delta\vec{F}^{max}$ are computed from current estimated state and the default boundary limits. For instance, if the last state estimation returns that $G_i = 100$ MW and the total generation capacity at the i^{th} bus is 120 MW, we have $\Delta G_i^{min} = 0 - 100 = -100$ MW and $\Delta G_i^{max} = 120 - 100 = 20$ MW. This constraint guarantees that the new dispatch $G_i + \Delta G_i$ is within the range of $[0, 120]$. When considering ramp speed, ΔG_i^{min} and ΔG_i^{max} will then be further constrained by the maximum/minimum ramp values. Similarly, if the estimated power flow exceeds the flow limits, then the corresponding transmission line is considered to be congested. Constraint (12.2c) is used to correct the flow rate, by moving the power flow at the next five minute to the range of $[-\vec{F}^{max}, \vec{F}^{max}]$, where $\vec{F}^{max}$ is the default real power flow limit on the transmission lines and the negative sign shows the potential reversed flow direction.

After solving (3.5), based on the optimization theory, we can obtain the corresponding optimal values of the dual variables λ, μ^- and μ^+. Note that dual variables are also known as Lagrangian multipliers, which are defined on each constraint, respectively. Their values at the solution point of the original linear optimization problem are known as the shadow prices. The LMP on bus i is defined as a linear combination of these shadow prices as follows:

$$LMP^E = \lambda \tag{3.6a}$$

$$LMP_i^C = \sum_j^M GSF_{j,i} \times (\mu_j^- - \mu_j^+), \tag{3.6b}$$

$$LMP_i^L = \lambda \times (-LF_i), \tag{3.6c}$$

$$LMP_i = LMP^E + LMP_i^C + LMP_i^L, \tag{3.6d}$$

where LMP^E, LMP^C and LMP^L are marginal energy price, marginal congestion price and marginal loss price, respectively; LF_i is the marginal loss factor at bus i. To compute this loss factor, we adopt the equation from [60], as follows:

$$LF_i = \sum_{j}^{M} 2 \times R_j \times GSF_{j,i} \times (\sum_{k}^{N} GSF_{j,k} \times (G_j - D_j)), \quad \forall i \in \mathbb{B} \tag{3.7}$$

where R_j represents the resistance at transmission line j.

3.2.7 Other Features

Aside from the above mentioned functions, we also enhanced the simulator with several other features to approach the real-world implementation.

Ramp Rate

Generation ramp rate is used to set a limitation on the changes that a power generator can achieve during a given time period. In the dispatch problem (3.5), without introducing a redundant constraint, ramp rate is implicitly implemented in the computation of $\Delta\vec{G}^{min}$ and $\Delta\vec{G}^{max}$. By default, we set ramping rate to 10%, which means the maximum generation change within five minutes is 10% of generator's capacity.

Generation Cost

We set multiple options for incremental generation cost $c_i(\Delta G_i)$ in (3.5). The naive option is the linear cost function $c_i = a_i * \Delta G_i$ which is seldomly used in practice but still popular in research. A more accurate model is the quadratic model where $c_i = a_i * ((\Delta G_i)^2 + 2G_i * \Delta G_i) + b_i * \Delta G_i$. Obviously, the computing complexity of the quadratic model is bigger than linear model and it is more sensitive to the minor changes in generation. Another option which is widely used in industry is the piece-wise linear model. Compared with the linear model, the parameter a_i changes with the current generation G_i. During our testing, the piece-wise linear model is the most often used.

Price-Responsive Load

The introduction of smart metering infrastructure enables two-way communication between power consumers and suppliers, leading to the thriving of demand-side management technologies [63]. One example is the price responsive load control which shifts the power consumption based on the electricity prices. The reshaped load curve will then affect the trend of real-time electricity prices. This process is described in [64] as a close-loop feedback control model. We used a similar model in the simulator to capture the price responsive behavior. Specifically, the real-time load $\vec{D}$ is the sum of the baseline demand and the price-responsive demand, which are denoted by $\vec{D}^{base}$ and $\vec{D}^{PR}$, respectively. The implementation of price-responsive load is summarized as follows:

1. Obtain the base LMP $\hat{\lambda}$, e.g., from the day-ahead market or using the LMP from the first time step of simulation.

2. Check if $\bar{\lambda}$, the average LMP over the last $\hat{n}$ steps, is greater than the sum of $\hat{\lambda}$ and a given threshold τ^{PR}. If not, terminate this process and use the five-minute load forecast as it is. Otherwise, continue the next steps.

3. (Optional) Given a predefined parameter γ, which quantifies the delay effect of the price responsive control or deferrable load, we update the time steps that requires load modification. For example, if the index of current time step is t, we mark $t + \gamma$ as the time step when price-responsive load will be applied. If current time is marked, continue; otherwise, terminate. In our simulator, the default value of γ is 1.

4. Given a predefined parameter β, which denotes the ratio of loads that cannot be affected by LMP, we have

$$\vec{D}^{base} = \beta \vec{D} \tag{3.8}$$

$$\vec{D}^{PR} = (1-\beta)\vec{D}\left(\frac{\bar{\lambda}}{\hat{\lambda}}\right)^{\kappa} \tag{3.9}$$

$$\approx (1-\beta)\vec{D} + \left(\frac{\kappa}{\hat{\lambda}}\right)(1-\beta)\vec{D}(\bar{\lambda}-\hat{\lambda}) \tag{3.10}$$

$$\vec{D} = \vec{D}^{base} + \vec{D}^{PR} \tag{3.11}$$

   where κ is a parameter used to control how fast the load can respond to the price change, and its default value is set as -0.8. Equation (3.10) is the first order approximation from Taylor expansion. By default, $\beta = 0.9$.

5. Use the adjusted demand $\vec{D}$ in IED, instead of the original forecast data.

3.3 Outage and Reserve Market

Equipment outages, e.g. line failure, generator trip or load loss, are common events in power grid operation. The scheduled outages are planned for maintenance or construction purposes and the unscheduled outages are mainly caused by equipment aging/overheating, relay tripping or extreme weather. To mitigate the impact of these outages, power grid regulation entities enforce a certain level of generation reserve for emergency control and quick recovery. This section introduces how the simulator incorporates dynamics caused by outages and reserve market. The flow chart in Figure 3.2 illustrates the switching between regular market and reserve market and their interactions with the outage events. The following subsections explain the details.

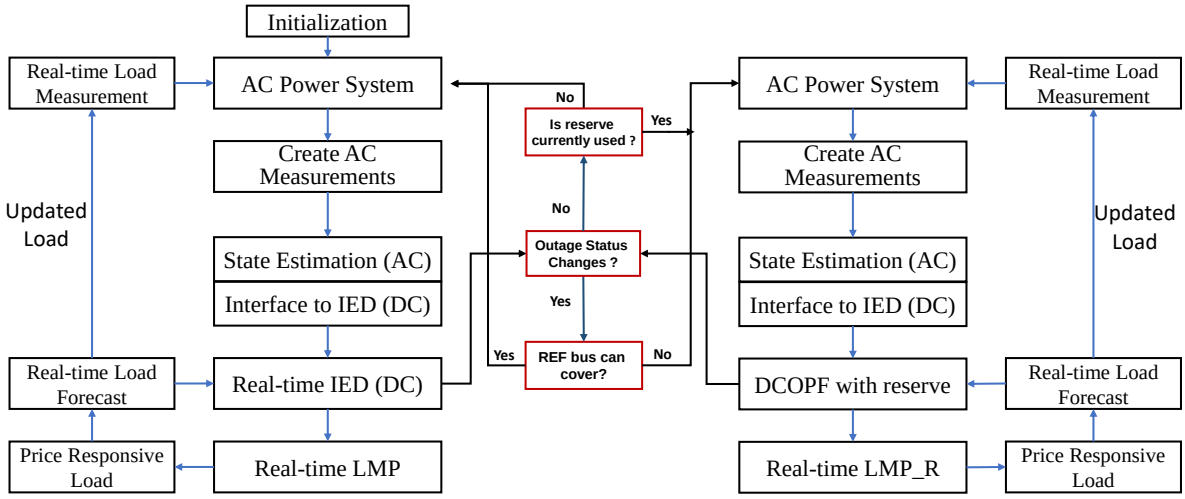


Figure 3.2: The electricity market simulator enhanced with outage management and reserve market.

3.3.1 Outage Management

To insert outages in the time series simulation, we first define the outage features in the parameter file, including the outage type (generator/line), location (which bus/line), start time, end time, and percentage of generation loss. During simulation, after each IED, the simulator will check if there is an outage event. If yes, the system will further verify if such outage can be absorbed by the current system condition using flexible generation on the reference bus. If yes, then the system continues to run on regular market with modified network topology or generation outputs. If no, the system will switch to reserve market and leverage the reserve capacity to cover the generation gap.

This process is marked as red blocks in Figure 3.2. Note we used two blocks to jointly judge if there is an outage at current time step. This is because we only defined the start and end time of the outage, corresponding to the *Outage Status Change?* block. During outage period, we use *Is reserve currently used?* block to check if simulator should continue with the reserve market or regular market.

3.3.2 Reserve Market

As stated above, under two conditions will the simulator switch to the reserve market: (1) there is currently an outage and (2) the reference bus does not have enough capacity to cover the generation loss. In reserve market, the simulator is almost a duplicate of the regular market except that the IED module is now a DCOPF module that co-optimizing both reserve and regular generation. The mathematical formulation is shown in 3.12.

$$\min_{G} \quad \sum_{i}^{N} c_i(G_i) + c_i^R(R_i) \tag{3.12a}$$

$$s.t \quad \sum_{i}^{N} R_i + G_i = \sum_{i}^{N} D_i \tag{3.12b}$$

$$F_j^{min} \leq \sum_{i}^{N} GSF_{k,i} \times (G_i + R_i - D_i) \leq F_j^{max},$$

$$\forall j \in \mathbb{L} \tag{3.12c}$$

$$G_i^{min} \leq G_i \leq G_i^{max}, \quad \forall i \in \mathbb{B} \tag{3.12d}$$

$$R_i^{min} \leq R_i \leq R_i^{max}, \quad \forall i \in \mathbb{B} \tag{3.12e}$$

where
$c_i(G_i)$: generation cost of regular generators at bus i;
$c_i^R(R_i)$: generation cost of reserve generators at bus i;
G_i: regular real power generation at bus i;
R_i: reserve real power generation at bus i;
F_j^{max}: maximum real power flow at line j;
F_j^{min}: minimum real power flow at line j;
G_i^{max}: maximum non-reserved real power generation at bus i;
G_i^{min}: minimum non-reserved real power generation at bus i.
R_i^{max}: maximum reserved real power generation at bus i;
R_i^{min}: minimum reserved real power generation at bus i.
λ: dual variable for constraints (3.12b).
$\vec{\nu}^-, \vec{\nu}^+$: dual variables for constraints (3.12c).

Using formulation (3.12), the definition of LMP can be carried over and expressed as (3.6a)-(3.6d). Note that when the reserve market is first activated, i.e. when an outage starts and reference bus fails to handle it, the regular generation G_i has already been dispatched by regular IED and will not join the co-optimization. This means we treat G_i in (3.12) as constant numbers in the first step and then treat them as variables in the following steps until the reserve is not needed for this outage event. Specifically, in the first step, we set both $\vec{G}^{max}$ and $\vec{G}^{min}$ to the same values decided by regular IED so as to fix $\vec{G}$ to the regular dispatch plan. In the following steps, similar to the regular IED, the boundary conditions $\vec{G}^{max}$ and $\vec{G}^{min}$ are determined by the ramp rate and maximal capacity. Once (3.12) is solved, we check if $\vec{R} \neq 0$, i.e. if reserves are still needed, and if not, we will move back to the regular market simulation, i.e., the left-hand-side of Figure 3.2.

In our simulator, we use the same ramp rate for the reserved real power generation, and their maximum capacity is 60% of the standard capacity, respectively. We set the cost $c_i^R(R_i)$ as a linear function of R_i, but with a higher cost rate. By default, we use the cost from the highest segment of the piece-wise linear function, multiplied by two.

3.4 Cyber Attacks

Among the three groups of attacks defined in Chapter 2, the False Data Injection Attacks (FDIA) and Physical Response Attacks are of special interest for dataset generation. This is because they are relatively more realistic and harmful to the system stability, compared to the Market Interface Attacks. The target of FDIA in this chapter is the SCADA system, responsible for data collection and remote control, and its interface to the central energy management system (EMS). By injecting falsified sensor data, the attacker aims to mislead the system control and dispatch functions to make non-optimal or even unstable decisions. In the Physical Response Attacks, the attacker tries to manipulate the behavior of physical devices in an aggregated manner such that the gathered impact can be sufficient enough to damage the system operation. One example is to shift the demand making it deviate from the forecast data and invalidate the generation plan. In our simulator we refer to such attack as Price Responsive Attack (PRA). A special case of the FDIA is Load Redistribution Attack (LRA) which modifies only the load forecast data rather than the metered measurements. This attack is more subtle and harder to detect. The implementation of the FDIA, PRA and LRA is demonstrated in Figure 3.3 on the baseline simulator model. The details are introduced in the following subsections.

3.4.1 LRA

LRA was first introduced by Yuan et al [30] followed by other researchers [65]. LRA tailors the load distribution in the target area strategically to tilt the dispatch results in favor of certain buses. LRA is hard to detect since it does not tamper with the well-protected

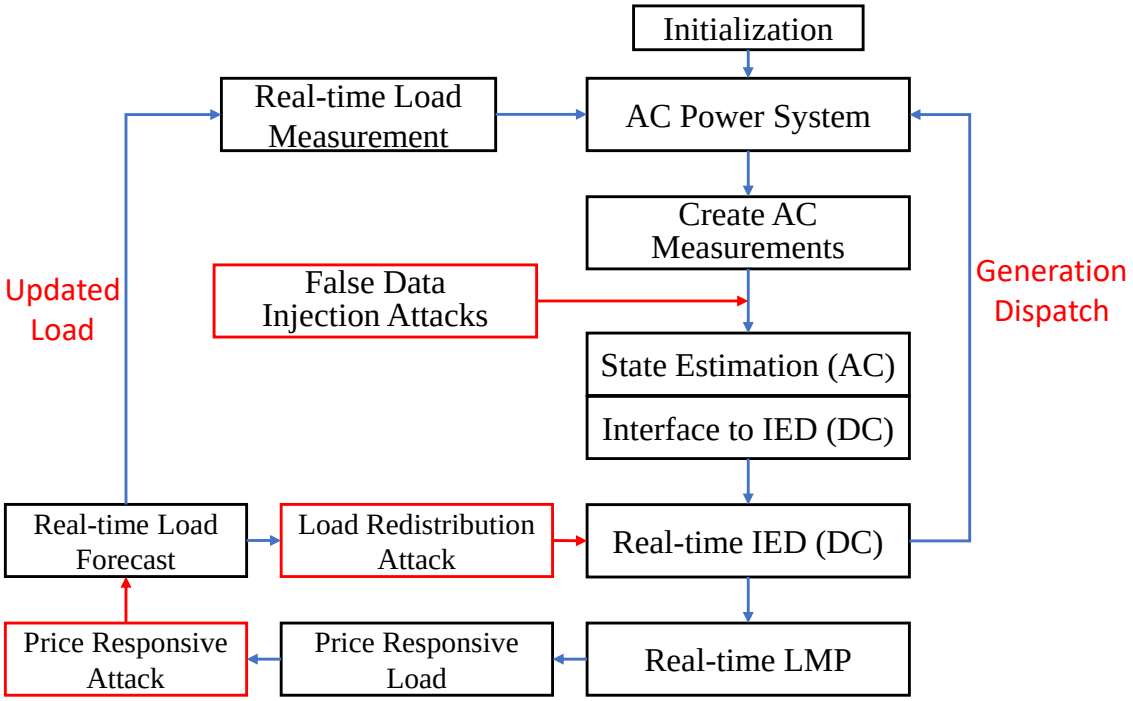


Figure 3.3: The electricity market simulator with different attacks.

generation buses nor affect the current state estimation results [30]. Meanwhile, LRA can undermine system stability by tricking ISOs to make erroneous dispatch decisions, which may overload certain transmission lines causing line outages.

In our simulation, LRA is implemented to redistribute the 5-min load forecast. It tries to increase the load prediction of the target buses so that their corresponding LMP will also be increased. In the meantime, to keep the total load and total generation unmodified in IED, the load forecasts of the non-target buses need to be decreased. Finding the optimal redistribution solution is out of scope for this chapter. Here, we simplified this process by decreasing the load at non-targeted buses proportionally to their original load forecasts. Additionally, to maximize the financial benefits, LRA is only activated during the critical hours, e.g., during the peak hours, when it is highly likely that LRA will increase the LMP at the target buses.

LRA is added by the following procedures:

1. Check if we have already applied enough attacks in a window of time period. If the number of existing attacks is greater than the maximum allowed attacks, terminate. By default, we allow no more than 12 attacks in the previous 2 hours.

2. Increase the load of the target bus i by the desired value ΔD_i^{LRA}, so that

$$D_i^{LRA} = D_i + \Delta D_i^{LRA}, \quad i \in \mathbb{B}^{target} \tag{3.13}$$

where $\mathbb{B}^{target}$ is the set of all the target buses; D_i is the actual next five-minute load forecast at bus i, while D_i^{LRA} is the falsified load forecast introduced by LRA.

3. Proportionally reduce the load at the non-target bus according to their next five-min load forecast. Note that we assume the adjusted load is greater than or equal to zero. That is

$$\begin{aligned} D_j^{LRA} =& D_j - D_j \times \frac{\sum_{i \in \mathbb{B}^{target}} \Delta D_i^{LRA}}{\sum_{k \in \mathbb{B}^{non-target}} D_k}, \\ & \forall j \in \mathbb{B}^{non-target} \end{aligned} \tag{3.14}$$

4. The forged load forecast is then passed to IED to obtain a wrong incremental dispatch plan $\Delta \vec{G}^{LRA}$ and manipulated LMP results.

5. Using the actual load forecast $\vec{D}$ and price responsive load to create the load measurement data. These data are fed into the *AC Power System* block, to get the power flow for next time step.

3.4.2 PRA

PRA is a type of load alternating attack [66], targeting on direct load control. It is inspired by the real-time pricing attacks [64], and MAnipulation of Demand attack (MAD) [67], which changes the load behaviors to damage the power grid. The motivation of our PRA is that the quick growth of smart grid foresees the wide usage of demand management technologies, which can reshape the load curves based on the real-time LMP information. Unlike the well-protected power grid infrastructure, the load controllers are located in the user end with much less security to defend against cyber attacks. The PRA is designed to inject false price signal to the load controllers so as to inverse the control logic, to use more power when LMP is high. Consequently, it may introduce additional line congestions at the peak hours.

Our implementation of PRA is summarized as follows:

1. (Optional) Check if we have already applied attacks in the previous steps. If the number of existing attacks is greater than the maximum allowed attacks in the given time period, terminate. For example, we can check if 5 continuous attacks happened during the last 2 hours.

2. Obtain the base LMP $\hat{\lambda}$, e.g., from the day-ahead market or using the LMP from the first time step of simulation.

3. Check if $\bar{\lambda}$, the average LMP over the last $\hat{n}$ steps is greater than the sum of $\hat{\lambda}$ and a given threshold τ^{PR}. If not, terminate this process and use the five-minute load forecast as it is. Otherwise, continue the next steps.

4. Given a predefined parameter β, which denotes the ratio of loads that cannot be affected by LMP, we have

$$\vec{D}^{base} = \beta \vec{D} \tag{3.15}$$

$$\vec{D}^{PR} \approx (1-\beta)\vec{D} + \left(\frac{-\kappa}{\hat{\lambda}}\right)(1-\beta)\vec{D}(\bar{\lambda} - \hat{\lambda}) \tag{3.16}$$

$$\vec{D}^{mod} = \vec{D}^{base} + \vec{D}^{PR} \tag{3.17}$$

5. Use the adjusted demand $\vec{D}^{mod}$ in IED, instead of the original forecast data.

Compared with the standard procedure used in creating price-responsive load in Section 3.2.7, in PRA, the controller logic is altered, by replacing the decreasing rate κ in (3.10) with increasing rate $-\kappa$ in (3.16).

Similar to LRA, PRA does not modify the sensor measurement data or the state estimation results, and hence it can bypass the SCADA detection system. However, unlike LRA, which aims to get a wrong dispatch without changing the real load in the power system, PRA actually changes the load behavior and such change is unexpected and unplanned in the power grid.

3.4.3 FDIA

FDIA is the third attack we implemented in the simulator. Unlike the other two attacks, FDIA needs to be well-designed to bypass BDD, and hence it is actively researched, both for design a successful realistic attack and for finding the defense countermeasure to protect the power system [29, 56, 65, 68, 69].

Based on the most practical state estimator and BDD scheme, as presented in Section 3.2.4, let $\vec{a}$, $\vec{z}_a = \vec{z} + \vec{a}$ and $\vec{\hat{z}}_a$ denote the false data injection vector, fake measurements and state estimation results from the fake measurement, respectively. Without carefully constructing the malicious data $\vec{a}$, the residual $\vec{r}_a = \vec{z}_a - \vec{\hat{z}}_a$ can break the residual test (3.4) and hence be easily detected by BDD.

In order to successfully hide the malicious attack, the attack vector $\vec{a}$ must satisfy the condition

$$\vec{a} = h(\vec{\hat{x}}_a) - h(\vec{\hat{x}}), \tag{3.18}$$

where $\vec{\hat{x}}_a$ is the estimated state under FDIA. This can be proven by the following equations:

$$\vec{r}_a = \vec{z}_a - \vec{\hat{z}}_a = \vec{z} + \vec{a} - h(\vec{\hat{x}}_a) \tag{3.19}$$
$$= \vec{z} + \vec{a} - (\vec{a} + h(\vec{\hat{x}})) \tag{3.20}$$
$$= \vec{z} - h(\vec{\hat{x}}) \tag{3.21}$$
$$= \vec{r}. \tag{3.22}$$

Therefore, if the original measurements and state estimation can bypass BDD, i.e., satisfies the condition (3.4), it implies $\|\vec{r}_a\|_2 \leq \tau$, too.

In order to construct $\vec{a}$ satisfying (12.1) in AC formulation, we follow a similar strategy proposed by [56], to minimize the changes in the states while lunching a successful attack. We can formulate this optimization as

$$\min_{\Delta\vec{V},\Delta\vec{\theta}} \quad \|\Delta\vec{V}\|_2^2 + \|\Delta\vec{\theta}\|_2^2 \tag{3.23a}$$
$$s.t \quad P_i^{inj}(\vec{V},\vec{\theta}) = P_i^{inj}(\vec{V}+\Delta\vec{V},\vec{\theta}+\Delta\vec{\theta}), \forall i \in \mathbb{B} \tag{3.23b}$$
$$F_{target}(\vec{V}+\Delta\vec{V},\vec{\theta}+\Delta\vec{\theta}) \geq F_{target}^{max}, \tag{3.23c}$$
$$\Delta V_i^{min} \leq \Delta V_i \leq \Delta V_i^{max}, \forall i \in \mathbb{B} \tag{3.23d}$$
$$\Delta\theta_i^{min} \leq \Delta\theta_i \leq \Delta\theta_i^{max}, \forall i \in \mathbb{B} \tag{3.23e}$$

where
$\Delta\vec{V}$: changes happened to the bus voltage;
$\Delta\vec{\theta}$: changes happened to the bus phase angle;
$\Delta\theta_i^{max}$: maximum changes in phase angle at bus i;
$\Delta\theta_i^{min}$: minimum changes in phase angle at line i;
ΔV_i^{max}: maximum changes in voltage magnitude at bus i;
ΔV_i^{min}: minimum changes in voltage magnitude at bus i.
$F_{target}(.)$: the real power flow on the targeted line;
$P_i^{inj}(.)$: the power injection at bus i; ;

The idea of this optimization problem is to find the minimum changes to the states, subject to (1) keeping the same power injection at all the buses, and (2) creating congestion at the targeted line. It is worth mentioning that this optimization problem is hard to solve, and there is no guarantee to find a solution. This is due to the fact, that if the flow on the targeted line is far away from its limit, there is no such solution that can make this line congested while keeping all the power injections unchanged. Therefore, we only apply this attack when the flow on the target line is close to its limit.

Solving this problem gives us the attack vectors to the state $c = (\Delta\vec{\hat{V}}, \Delta\vec{\hat{\theta}})$. We can then

get the full attack vector by setting

$$\vec{a} = h(\vec{\hat{x}} + \vec{c}) - h(\vec{\hat{x}}) \tag{3.24}$$

This FDIA can successfully push the target line flow to the limit and make it look congested in the state estimate. Therefore, the following IED, which uses this fake state estimation, cannot assign any more flow to the target line, and has to shift the flow onto other routes if needed. Note that the incremental results from IED are then added to the real states without FDIA. Consequently, this wrong dispatch plan increases the chance that some non-target line can get congested. However, we need to highlight that it is still possible that FDIA fails to create any impact. For example, if there is a big increase in the load in the next five minutes, the IED may have to use full capacity of the targeted line, regardless of the FDIA. On the other hand, if there is a big decrease in the load, the IED will remove the fake congestion on the target line introduced by FDIA. As a result, this FDIA fails to create any impact in the dispatch results, even though it is well-structured and can create a fake congestion.

Our implementation of FDIA is summarized as follows:

1. (Optional) Given a time window. Only apply FDIA if current time-stamp is within the given window. Otherwise, terminate.

2. Given a target line for FDIA, where the attackers would like to create a fake congestion in the corresponding state estimation results.

3. Check if the flow rate on this target line is close to the limit. In our simulator, we have two triggers: (1)$F_{target} >= \ell_r * F_{target}^{max}$ and (2) $F_{target} >= F_{target}^{max} - \ell_c$. If one trigger is denied, terminate. (In our test case, $\ell_r = 0.75$ and $\ell_c = 100$.)

4. Compute state estimation $\vec{\hat{x}}$ from the real measurements $\vec{z}$. See Section 3.2.4 for details.

5. Solve the optimization problem (12.2) to get $\vec{c}$ and fake state estimate $\vec{\hat{x}}_a = \vec{\hat{x}} + \vec{c}$. If it fails to converge, terminate.

6. Create attack $\vec{a}$ by equation (3.24), and get the false measurement $\vec{z}_a = \vec{z} + \vec{a}$.

7. (Optional) Get fake state estimate from false measurement $\vec{z}_a$ according to Section 3.2.4. Verify it with $\vec{\hat{x}}_a$. If the error is out of the given tolerance, terminate.

8. (Optional) Continue to the next module IED to compute two dispatch plans. One is with FDIA, and the other is from the original state estimation without FDIA.

9. (Optional) Check if FDIA can alter the congestion pattern. If yes, the simulation found a successful FDIA and use it in the current step; otherwise, abort FDIA and use the real measurements, state estimation and corresponding IED in current step.

Since the state estimation uses a Newton-type iterative method which only guarantees local convergence, we can use the optional step 6 for verification to guarantee the solution of (12.2) is in a local region of interest. We mark step 7 and 8 as optional as well, since these two steps require using the IED for post-fact verification to guarantee a successful FDIA that can change the congestion pattern as designed.

3.5 Numerical Experiment

We tested the simulator on the PJM 5-bus system, IEEE 14-bus, 39-bus and 118-bus system. For easy interpretation and reasonable complexity, we explained in detail the results on the IEEE 39-bus system, shown in Figure 3.4. We used 21 load forecast data downloaded from PJM Data Miner 2 [70]. The day-ahead hourly load forecast and 5-minute real-time load forecast from Oct. 17th to Oct. 24th 2019 were used in initialization and IED, respectively. The aim of numerical experiments is to demonstrate the system responses with/without cyber attacks and with/without outages. The dataset generated will be used in Task 4 for algorithm development and evaluation. The signatures of normal/abnormal events will be used for feature selection and root-cause diagnosis.

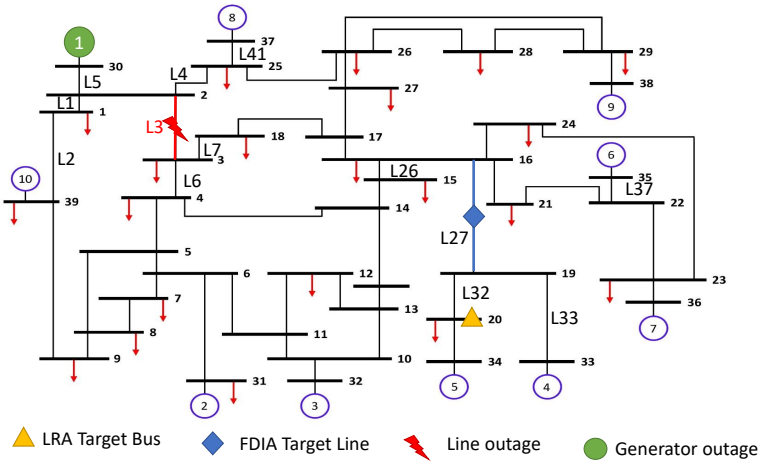


Figure 3.4: IEEE 39-bus system.

Note that some slight modifications were added in the MATPOWER case file "case39.mat". These include:

1. Using different generation cost parameters $a_i G_i^2 + b_i G_i + c_i$ for bus i. Instead of using the default uniform cost parameter where $a = 0.01$, $b = 0.3$ and $c = 0.2$, we used parameters in Table 3.2. By setting different cost profile, the IED can have a unique solution for each generator.

Gen ID	a	b	c
1	0.1	30	2
2	1	300	2
3	0.1	35	2
4	0.1	40	2
5	0.1	50	2
6	0.1	60	2
7	0.1	70	2
8	0.1	80	2
9	0.1	90	2
10	0.1	100	2

Table 3.2: Generation cost for IEEE 39-bus system

2. Setting different flow limit to lines. Instead of using the default values, we first removed all the flow limits and did an N-1 contingency analysis. From each contingency case, we calculated the average flow on each line over the simulation period. Then we used the maximum average flow over all the contingency analysis, as the flow limit in IEEE 39-bus system. These values are provided in Table 3.3.

We focused on the 24-hour simulation, which has 288 time steps all together. The power generation plan over all the power generators is presented in Figure 3.5, while the corresponding 24-hour LMP is presented in Figure 3.6. We can observe that the total demand varies from 4200 MW to 5500 MW.

In the following subsections, we first illustrate the system responses under three cyber attacks, respectively. We then show system operation conditions under line and generator outages. Finally, we test scenarios when both attack and outage happen simultaneously. For each test case, we analyze the results and explain in detail the impact and signatures.

3.5.1 LRA

For LRA, we used bus 20 as the target bus and set the attack period as 5 AM to 7 AM, i.e. the morning demand peak time. During LRA, we increased the demand at bus 20 by 50MW, roughly 1% of the total demand. The LMP with/without LRA are compared in Figure 3.7, which zooms in the period from 4 AM to 9 AM, to better present the LMP behaviors under LRA.

We can see LMP of bus 20 starts increasing earlier than the one without LRA. This is because the attacked dispatch shifts the line congestion from line 27 to line 33 when LRA is applied. For these two lines, we plot the flow rate from DC dispatch results in Figure 3.8 where the black curve *Flow* represents the flow without LRA; the blue dash curve *Flow_pred_LRA* represents the dispatch results using the fake load profile and the green

Line ID	Flow Limit	Line ID	Flow Limit
1	597.61	24	534.43
2	511.16	25	803.51
3	787.59	26	804.11
4	329.80	27	633.19
5	836.53	28	709.40
6	638.97	29	677.94
7	450.60	30	577.81
8	643.61	31	298.85
9	751.90	32	304.04
10	572.62	33	652.00
11	723.80	34	508.00
12	719.37	35	941.57
13	751.20	36	594.70
14	425.19	37	647.94
15	544.37	38	940.05
16	580.80	39	571.29
17	566.55	40	454.28
18	719.77	41	538.71
19	676.22	42	364.05
20	725.00	43	162.62
21	191.23	44	191.93
22	198.51	45	315.16
23	667.86	46	727.90

Table 3.3: Flow limit for IEEE 39-bus system

dash curve *Flow_LRA* represents the dispatch results using the real load profile. When an LRA happened, operators use the falsified load in IED and expect to see the flow behave as the blue curve. However, the real load will adjust the dispatch result and have the flow acting as the green curve, and hence overheat the transmission line.

We also present flow rates for line 26 and line 32 in Figure 3.9. From these figures, we observe that LRA can successfully redistribute the power flow to overload the transmission line 27, and also make line 33 get congested earlier. However, it does not affect the flow rate on line 32. This is because the evolving of power flow is driven by the demand. Since the actual power demand doesn't change under LRA and line 34 is always congested, line 32 is the only route to deliver the required power to bus 20 and hence LRA cannot change the congestion pattern of line 32.

We observe that the impact of LRA disappears after 6 time-steps though we applied 12 continuous attacks. The reason is that the load at bus 20 is increasing after 5 AM. When

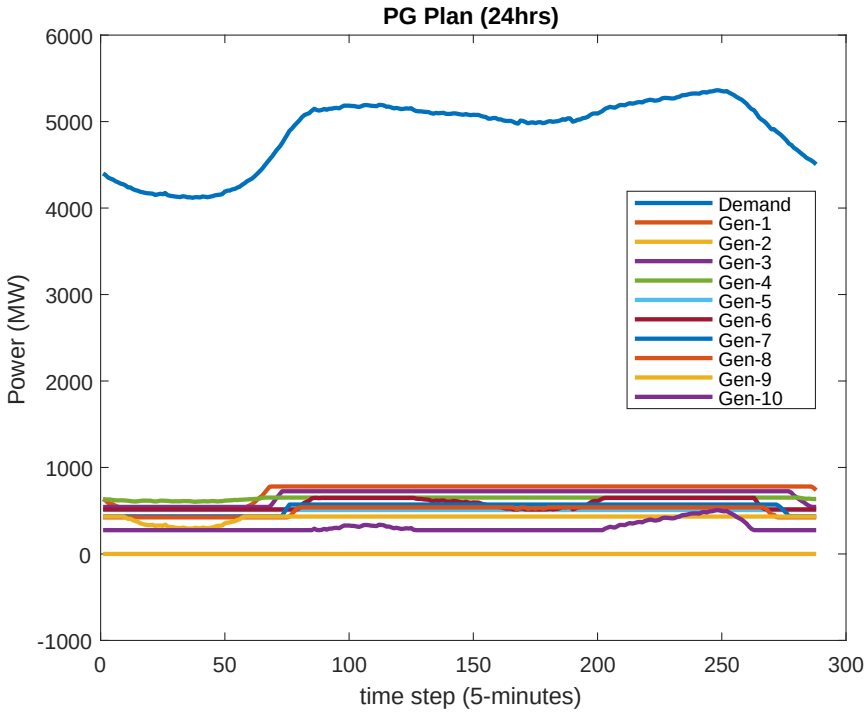


Figure 3.5: Power generation and total demand for a 24-hour simulation of IEEE 39-bus system

this load reaches a certain value, the local generators need to deliver more power to bus 20 instead of delivering cheap power to the rest of network through line 27. As a result, the congestion on line 27 is cleared. This shows that the impact of LRA depends on the trend of actual load.

We also observe that there is a big jump in flow rate from Figure 3.8 and Figure 3.9, whenever we activate or deactivate LRA. However, we can observe similar behavior in flow rate during the 24 hour simulation without attacks, from Figure 3.10. Therefore, LRA is a stealthy attack and can be hidden in the system dynamics.

3.5.2 PRA

To perform a PRA, we set $\tau^{PR} = 20$, $\hat{n} = 6$ and $\gamma = 1$. That is, when the average LMP over the last 30 minutes (6 steps) is above the sum of initial LMP and τ^{PR}, we mark the next step as the time price-responsive load is activated.

We present the total power demand with/without PRA in Figure (3.11a) and their corresponding maximum LMP over the 39 buses in Figure (3.11b). From the red curve, which represents the case without PRA, we can observe that LMP reaches the peak value around

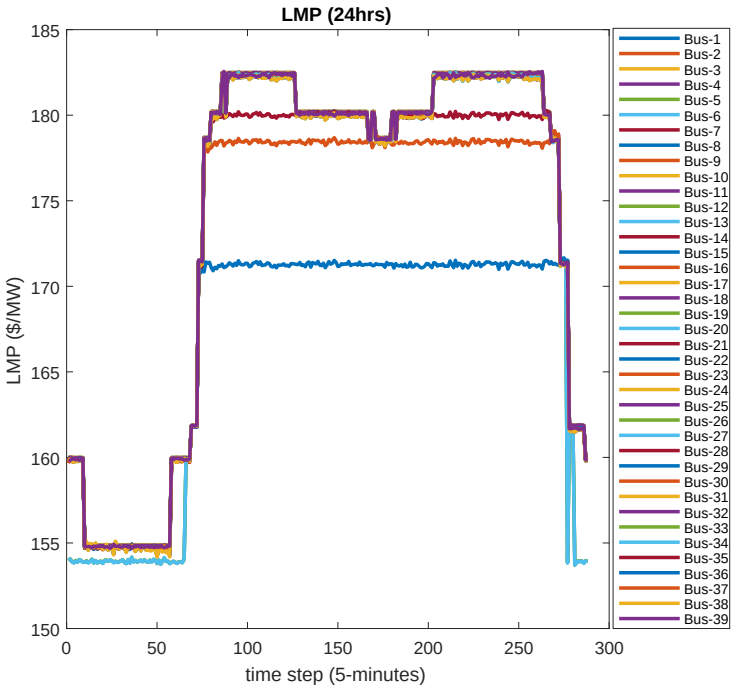


Figure 3.6: LMP for a 24-hour simulation of IEEE 39-bus system

$198 due to the increase in demand. This figure also shows that whenever LMP is above around $180, the load curves start to differ. The load in blue dash line is higher than the load in red since PRA increased the load by reversing the control logic of demand response program. When the price-responsive load is not activated, the load is set back to the default value $\vec{D}$ obtained from the five-minute load forecast.

Unlike LRA, PRA actually changes the load behavior and the entire dispatch results, including flow rates on the transmission lines and power generation. Thus, it is able to manipulate both energy cost LMP_E and congestion cost LMP_C. In this test case, the given PRA successfully changes the congestion pattern in line 37 and line 41, shown in Figure 3.12. Additionally, the increases in flow occurred when attack is activated or deactivated are slightly smaller than those caused by LRA. This is because in PRA the same load is used in solving IED and creating measurements, i.e. no fake load data are involved.

3.5.3 FDIA

For FDIA, we used line 27 as the target line where we created fake congestion by injecting false data into measurement. We applied 9 successful FDIA during the period between 4

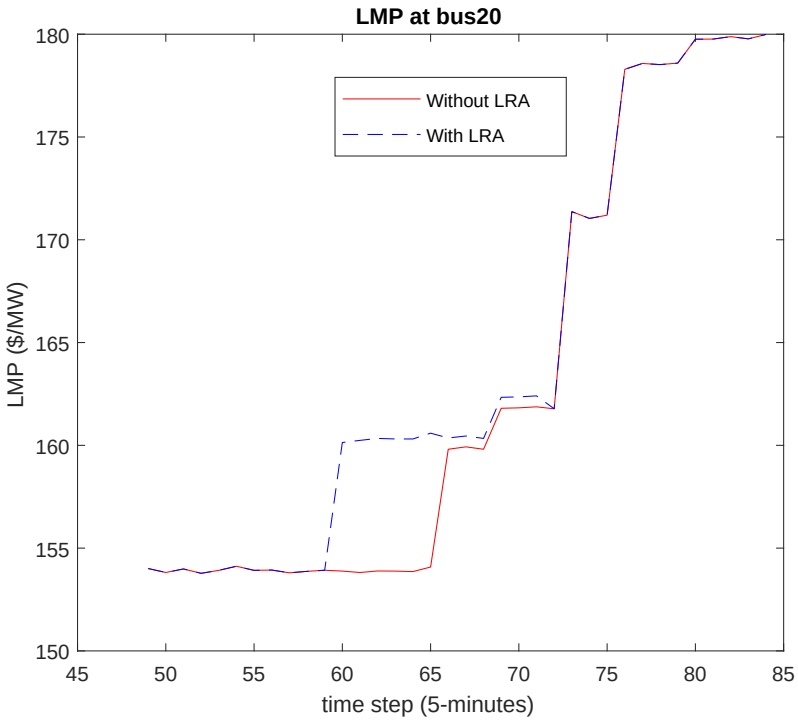


Figure 3.7: Comparison between LMP with/without LRA at bus 20.

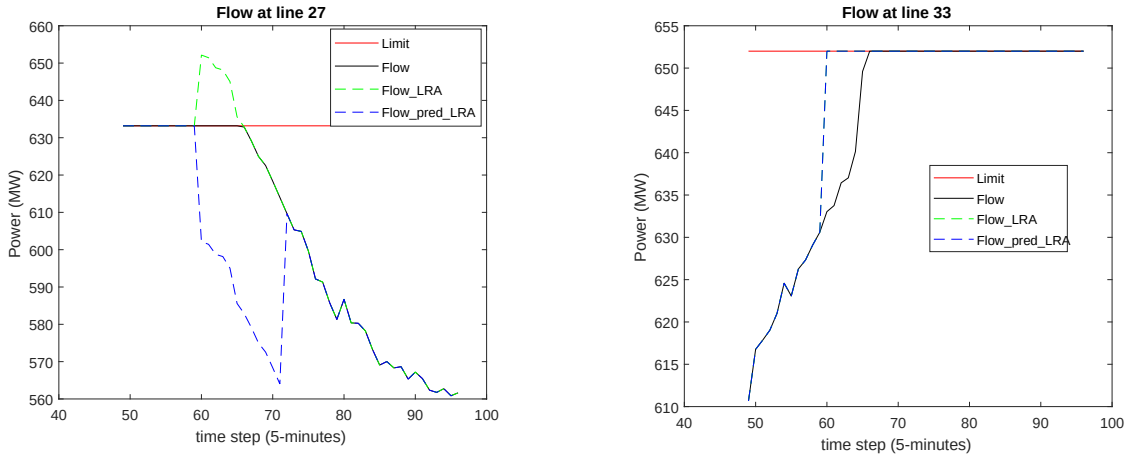


Figure 3.8: Power flow on line 27 and line 33 with/without LRA.

PM and 9 PM, scattered at time steps 192, 195, 214, 223, 224, 231, 233, 236 and 246. The measurements and state estimation results for line 27 are presented in Figure 3.13, where we can see 9 attacks successfully move the target flow above the limit. Therefore, in the

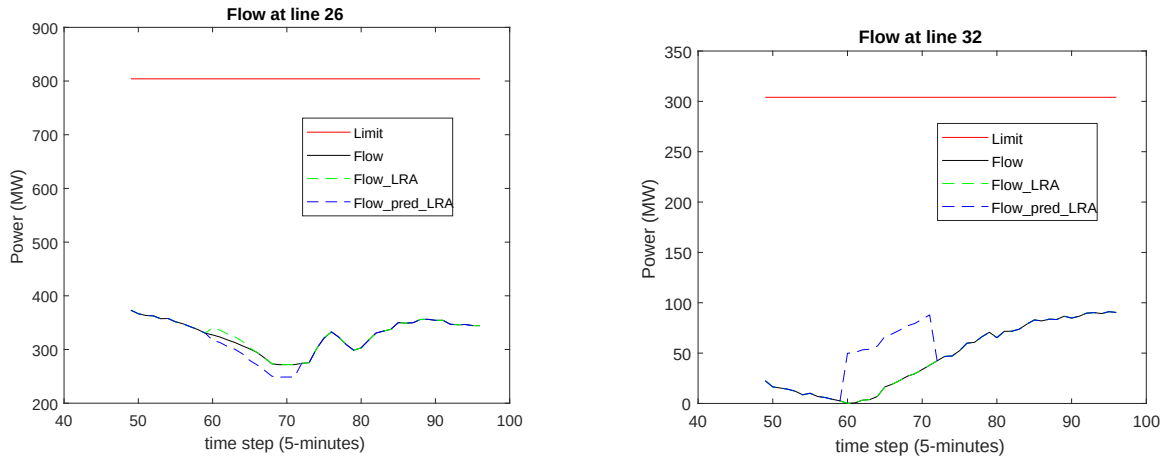


Figure 3.9: Power flow on line 26 and line 32 with/without LRA.

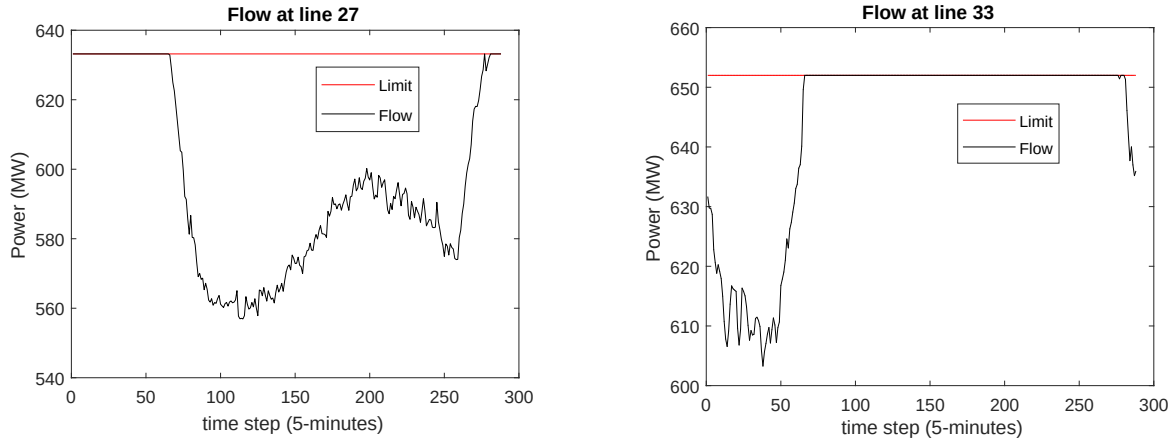
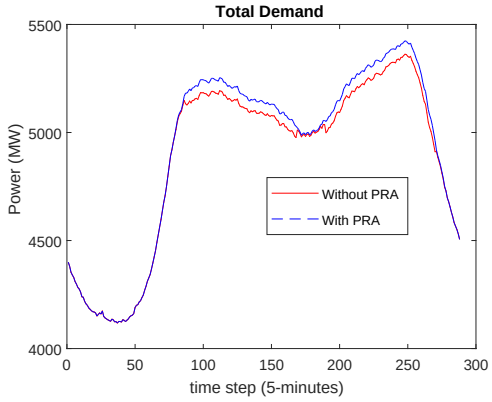


Figure 3.10: 24-hour power flow on line 27 and line 33 without attack.

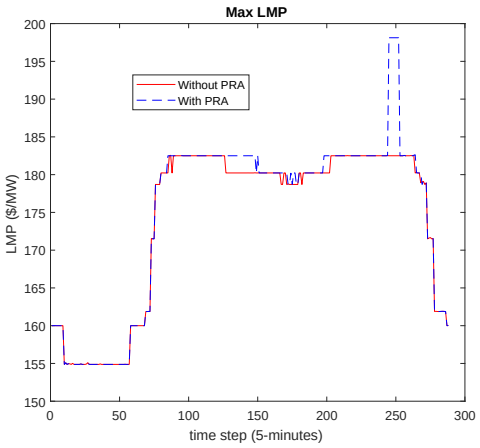
following IED module, no more incremental flow can be assigned to line 27. In fact, the solution from IED will remove the over-limit flow to make sure the new dispatch is under the flow limit. Consequently, power flows on other transmission lines will be increased, in order to deliver the necessary power to supply the demands in the network.

Due to the optional step 8 in Section 3.4.3, these 9 attacks can successfully change the congestion pattern after dispatch, resulting in some spikes in the LMP curve shown in Figure 3.14. Note that the changes are not dramatic, compared with the other two attacks. This is expected since the objective of nonlinear optimization problem (12.2) is to minimize the change of the states. Consequently, even though the attacks are successful, the changes in state estimation are subtle.

Without verifying through step 8 in Section 3.4.3, FDIA cannot guarantee the change of

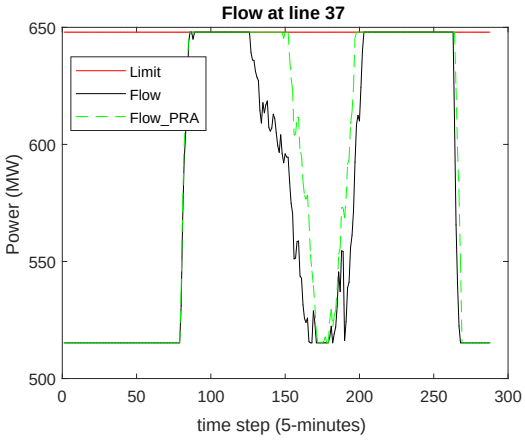


(a) Total power demand



(b) Max LMP over the 39 buses

Figure 3.11: Total demand and maximum LMP with/without PRA.



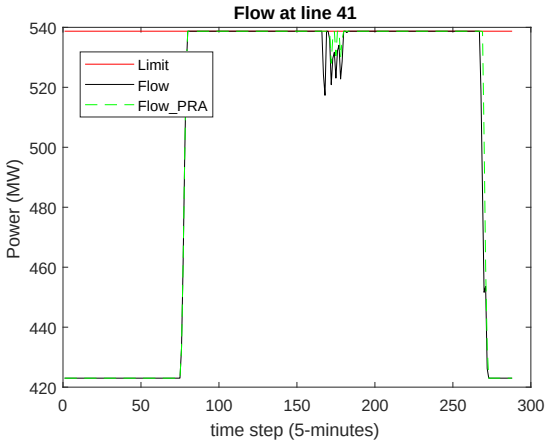
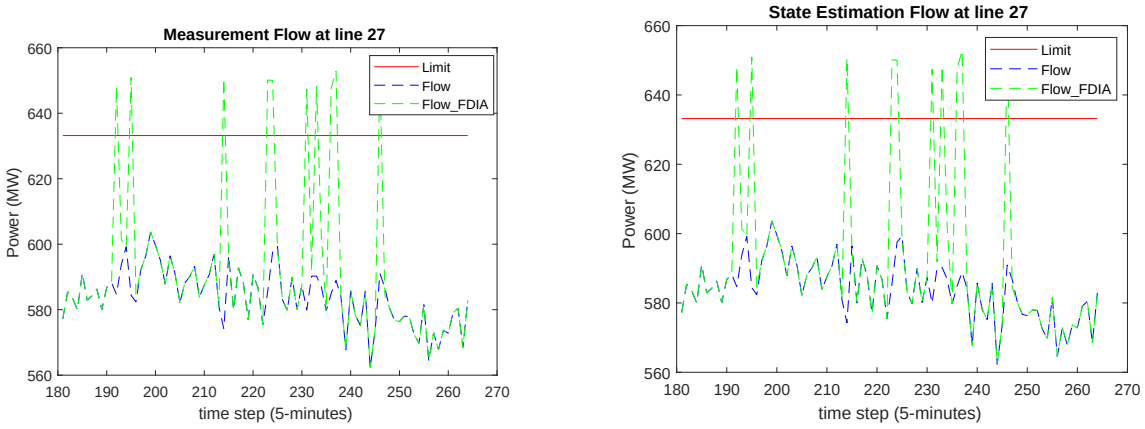


Figure 3.12: 24-hour power flow on line 37 and line 41 with/without PRA.

congestion patterns, since we cannot foresee the dispatch results for the next step. If we add more constraints into this optimization problem, it can easily become an over-determined problem as discussed in [56]. For example, asking for 2 specific congested lines may make the problem infeasible. The attacker also needs to know more real-time information about the power grid, in order to create an attack that can bypass BDD and also create a big impact.

Last but not least, through all three attacks, we notice that the impact of attacks can be quickly absorbed by the power system. In other words, once the attack is discontinued, the power system can rapidly recover to the normal operation status in a couple of steps. This means to achieve a big gain the attacks must either (1) have a huge impact in the current step; or (2) continuously attack the system for a while. In the following cases, we will show



(a) Flow from measurements.

(b) Flow from state estimation.

Figure 3.13: Measurement and state estimation at line 27 with/without FDIA.

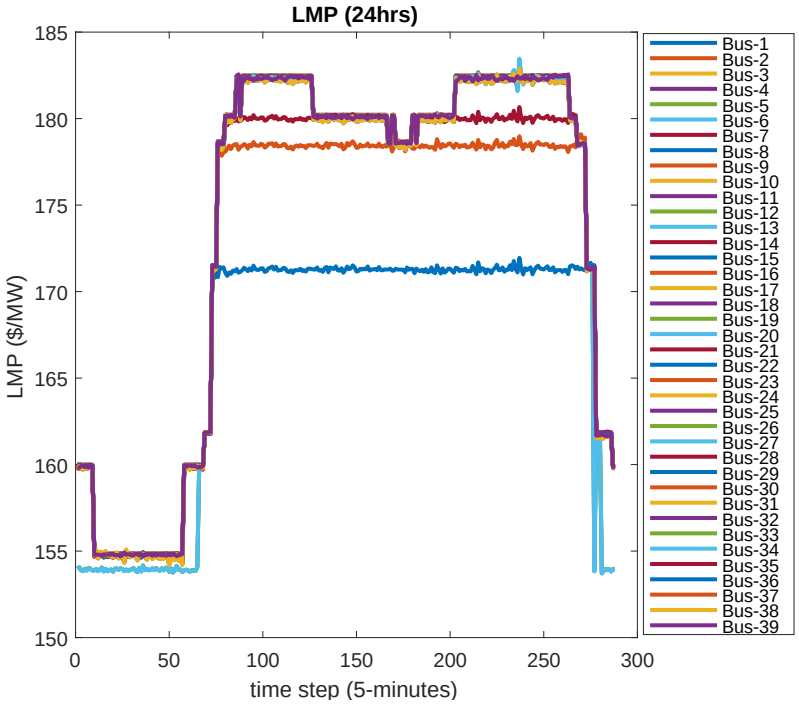


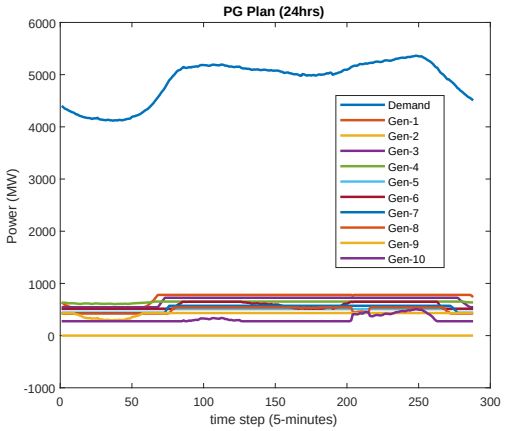
Figure 3.14: LMP with FDIA

how cyber attacks can damage the power grid during outages.

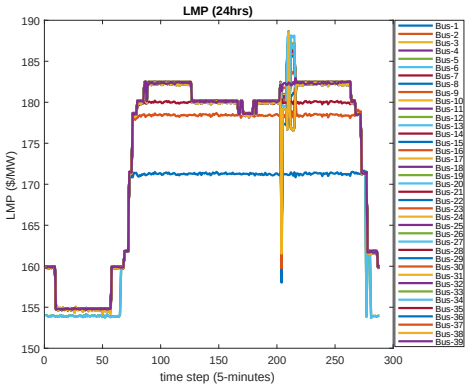
3.5.4 Line Outage

In this test case, we assume there is an unexpected line outage on line 3, from time 17:55 to 18:55. Note that when the line outage happens, the topology of power grid changes and hence we need to recompute a new GSF matrix and use it in the IED formulation (3.5).

We present the power generation in Figure 3.15a and LMP in Figure 3.15b during line 3 outage. The line flow on line 3, and the lines nearby are presented in Figure 3.16. From these figures, we can see the flow on line 3 suddenly decreases to 0, and hence there is a flow surge in line 7, in order to cover the demand on bus 3.



(a) Power generation with line outage.



(b) LMP with line outage.

Figure 3.15: IEEE 39-bus system with line outage at line 3.

If we switch on the PRA during line 3 outage, we can see a big surge in LMP from Figure 3.17. This is because the congestion is removed from line 41, while three new congestions are introduced at line 37, 40 and 42, when the line outage happens.

If we apply one FDIA during line 3 outage, at time step 208, we can observe the maximum LMP changes slightly, from Figure 3.18. However, without line outage, the FDIA will not impact the system at all, shown as the black dot curve.

To test the impact of LRA during line outage, we tested LRA with different amounts of load shift. We can observe the larger load shifts from LRA, the more LMP deviates from the no-attack case in Figure 3.19, where the curves *LMP_Lout*, *LMP_Lout_LRA_50*, *LMP_Lout_LRA_100* and *LMP_Lout_LRA_200* represent the LMP without attack and with 50/100/200 MW load shift, respectively.

3.5.5 Generation outage

In the last cases, we tested the most complex scenarios, where unexpected generation outage and reserve market are activated and different attacks are applied simultaneously.

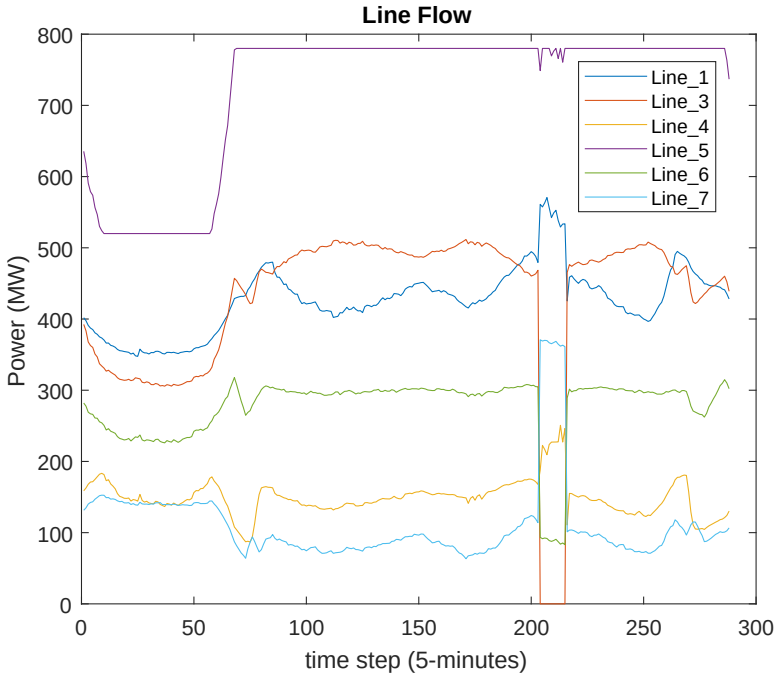
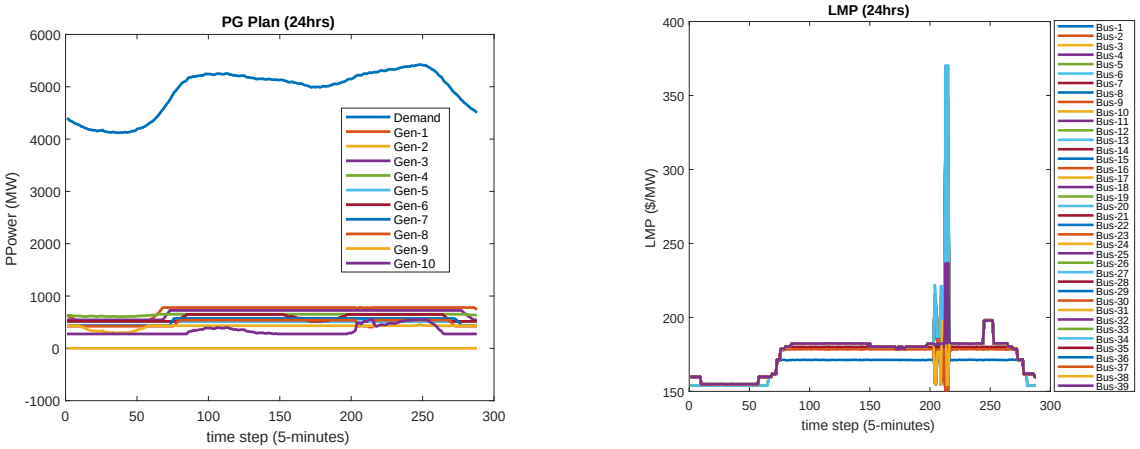


Figure 3.16: Power flow with line outage.



(a) Power generation with line outage and PRA. (b) LMP with line outage and PRA.

Figure 3.17: IEEE 39-bus system with line outage and PRA.

For comparison, we started with only one generation outage, which occurs on generator 1 (G1) at bus 30, from 12:00 PM to 12:30 PM. Without outage, G1's power generation during this period is fixed to 780 MW. We consider three different scenarios: (1) the power

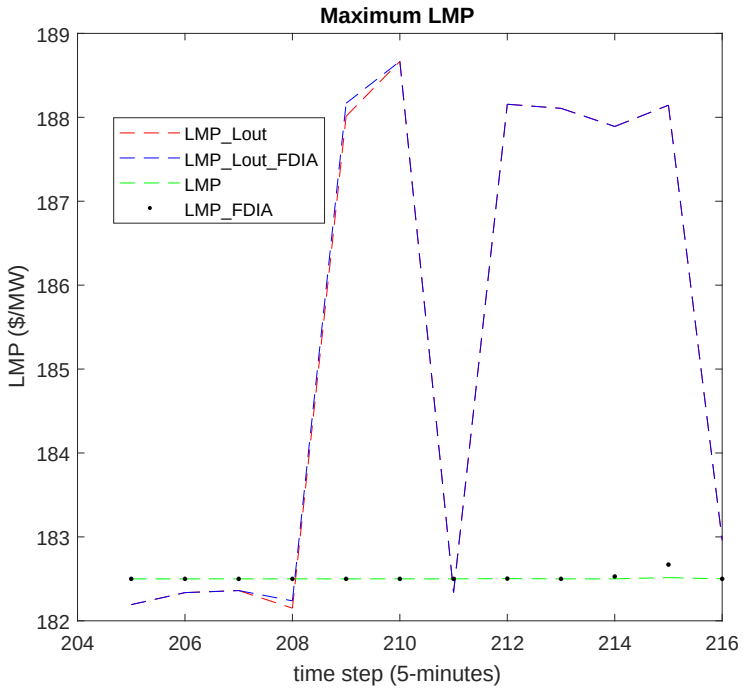


Figure 3.18: Max LMP with/without line outage and with/without FDIA.

generation reduces to 728 MW (93% of the original value); (2) the power generation reduces to 572 MW (73% of the original value); and (3) the power generation reduces to 416 MW (53% of the original value). The changes in power generation are demonstrated in Figure 3.20.

In the first scenario shown as Figure 3.20a, we observe that the outage can be covered by the reference bus since the generation loss is small. Therefore, no reserve generation is required. From the next time step, new dispatch can reallocate the regular power generation to balance the system. As a result, generator 6 and generator 3 need to generate more during G1 outage. When the outage is cleared, G1 returns to its default generation value, 780 MW.

In the other two scenarios, we can see the reserve is required in Figure 3.20b and Figure 3.20c, since the reference bus cannot cover these big losses due to its ramp rate limit. However, in the 2nd scenario, the system is able to return to the regular market in the next step, where IED can find a feasible dispatch plan without using the reserves. In contrast, in the 3rd scenario, the loss is too large and it cannot be resolved in one step due to the ramp limit. Hence it requires the usage of reserves for 2 sequential time steps, until the regular generators can replace the reserves. The corresponding LMP is shown in Figure 3.21. Note that due to the surge in LMP when reserve is required, the price-responsive load is activated and hence we can observe a load decrease during this period in Figure 3.21.

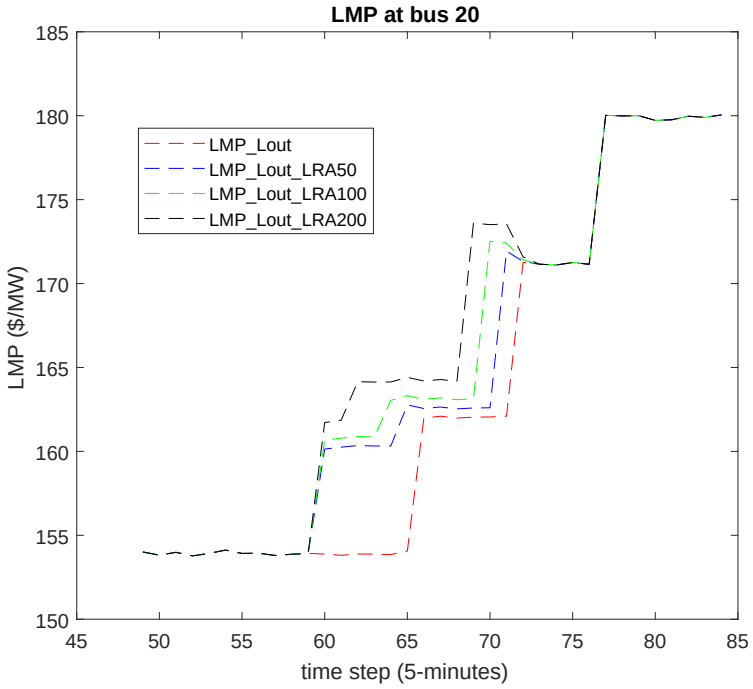


Figure 3.19: LMP under different LRA with line outage.

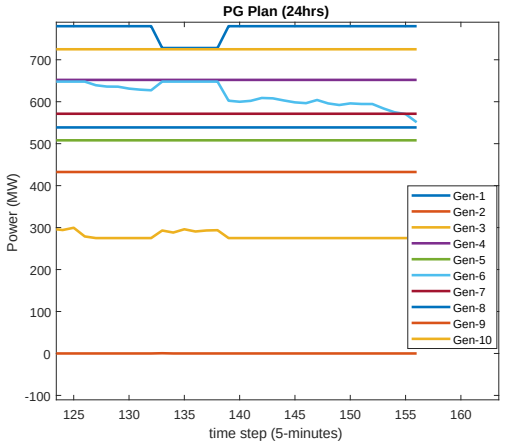
If we apply PRA attacks during the generation outages in scenario 2 and scenario 3, we can see that the LMP reaches roughly $500 during the outage in scenario 2, from Figure 3.22. For scenario 3, the PRA and generation outage overload the power system, and hence there is no dispatch plan that can satisfy the demand. This case demonstrates cyber attacks can destabilize power grids when applied at critical time.

3.5.6 Summary

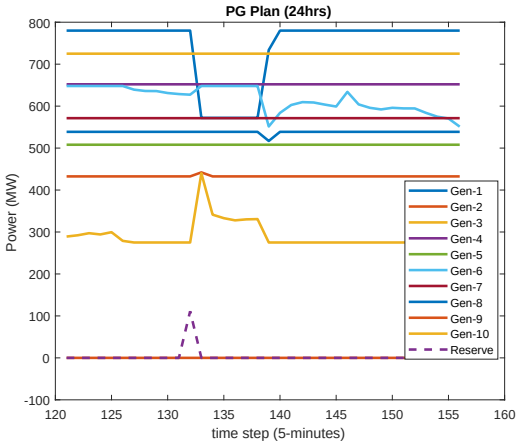
The major findings from the numerical experiments are summarized as follows:

- Signatures of LRA are:
  (1) misalignment between the load measurement trend and LMP dynamics (congestion patterns);
  (2) misalignment between the load measurement trend and the predicted load trend from solving the dispatch problem;
  (3) overload on unexpected transmission lines due to erroneous dispatch;
  (4) more impact at peak load hours or during outages.

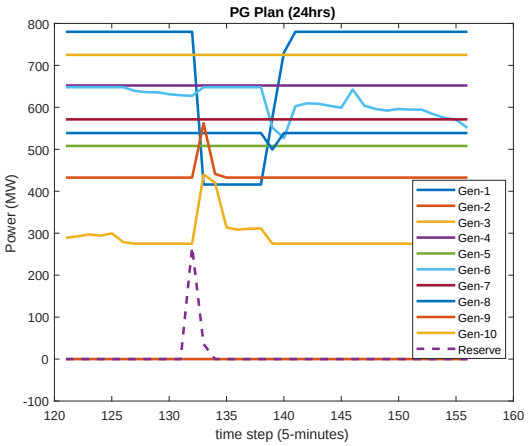
- Signatures of PRA are:
  (1) no reaction with meter measurement and state estimator, thus harder to detect;



(a) Maximum G1 is 728 MW during the outage.



(b) Maximum G1 is 572 MW during the outage.



(c) Maximum G1 is 416 MW during the outage.

Figure 3.20: IEEE 39-bus system with generation outage at G1.

(2) misalignment between the historical load trend and load measurement;
(3) delay effect due to demand response time;
(4) more impact at peak load hours or during outages.

- Signatures of FDIA are:
  (1) more stealthy thus harder to detect;
  (2) overload on transmission lines due to erroneous dispatch;
  (3) more accurate in creating desirable congestions in measurements and state estimation results;
  (4) unexpected change in LMP and damage to the power system.

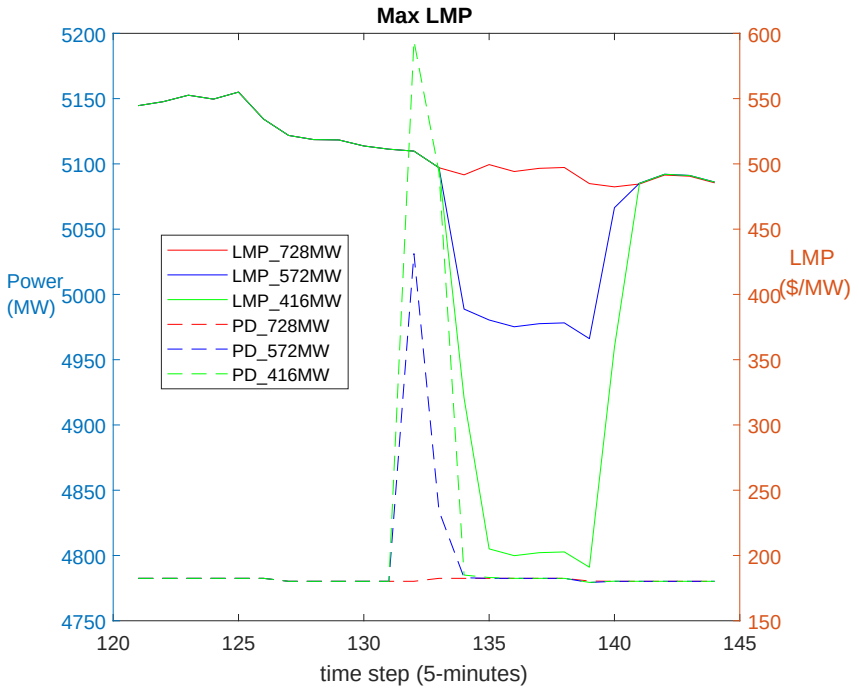
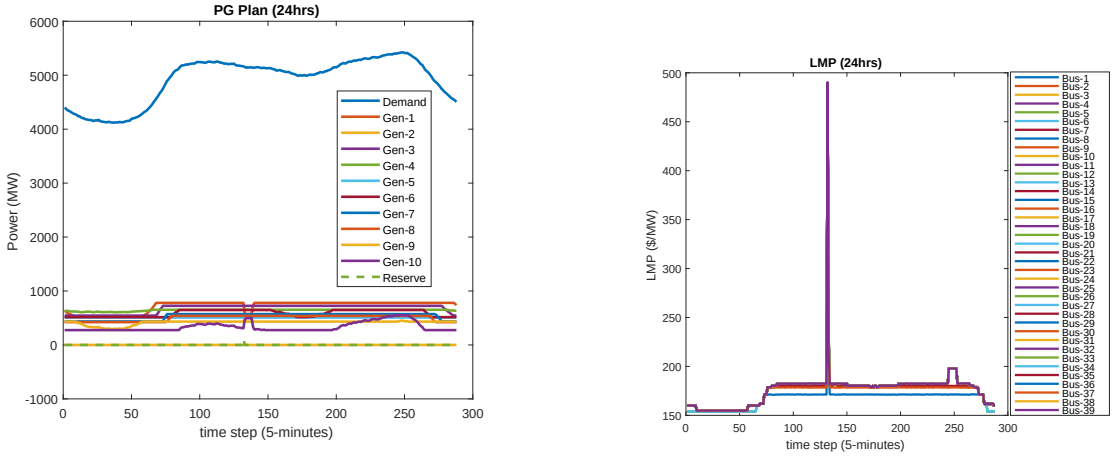


Figure 3.21: Total real demand (PD) and LMP with generation outages.



(a) Power generation and demand with generation outage and maximum G1 is 572 MW..

(b) LMP with generation outage and maximum G1 is 572 MW.

Figure 3.22: IEEE 39-bus system with generation outage and PRA.

- Signatures of line outages are:
  (1) sudden dip of power flow on one transmission line;

(2) sudden dramatic change in LMP (congestion patterns and different generation shift factor);
(3) flow pattern changes dramatically, e.g., flow changes direction.

- Signatures of generation outages are:
(1) reserve market response during severe generation loss;
(2) sharp spikes in LMP;
(3) misalignment between the historical real generation trend and the run-time real generation trend.

- Cyber attack creates bigger impact on power grids when applied (1) at critical time; or 2) during outages.

- The attack impact can be accumulated to destabilize the system when cyber attack is applied continuously.

3.6 Conclusion

This chapter presented results from the effort on creating realistic simulation dataset and deriving event signatures. An electricity market simulator was built with multiple innovations in supporting time series simulation and cyber attack implementation. Using this simulator, the team tested scenarios in long/short terms with/without cyber attacks and with/without outages. It is demonstrated that cyber attacks can be easily disguised in system dynamics and early detection of stealthy cyber attacks is essential in preventing further damage to the power grid.

Chapter 4

WISP Algorithms: Anomaly Detection

Information and communication technologies have been widely used in smart grid applications for efficient operation. However, these technologies are vulnerable to malicious cyber attacks, which may lead to severe reliability and economic issues. Recently, a variety of data-driven anomaly detection approaches have been explored to detect potential cyber attacks in smart grids. In this chapter, we present research on the electricity market data aiming to identify anomalies from the locational marginal prices (LMPs) and provide a new indicator for potential cyber attacks in power grids. Specifically, a novel data-driven anomaly detection framework is proposed for electricity market, which consists of three major components: (1) real-time point-wise anomaly detection, (2) real-time locational anomaly detection and (3) price spike anomaly detection. The following sections will introduce the algorithms and evaluation results on multiple data sets. Specifically, Section 4.1 presents the probabilistic anomaly detection framework focusing on an optimization based algorithm. Section 4.2 presents deterministic anomaly detection framework with novel threshold optimization method. Section 4.3 presents ensemble results of deterministic algorithms on the simulation dataset. Section 4.4 presents locational detection algorithms tested on PJM dataset. Section 4.5 presents locational detection algorithms tested on the simulation dataset. Section 4.6 presents price spike detection algorithms evaluated on PJM and ISO-NE datasets. Section 4.7 concludes the chapter.

4.1 Real-time Point-wise Anomaly Detection - Part I Probabilistic Methods

4.1.1 Introduction

The U.S. power grid is a complex cyber-physical system incorporating a vast volume of distributed devices, which by nature results in a large attack surface. Malicious attackers

can compromise the power devices, communication and control facilities or market interfaces, leading to local outages, equipment damage, grid instabilities or individual financial gains. A promising cyber defense approach, which is non-intrusive to the operational system and adds additional protection is physical response based anomaly detection. For cyber-physical systems, evaluating physical performance from sensor data is a common practice, but using these data to detect cyber attacks is under-developed. A few anomaly detection technologies have been presented in the literature with power systems applications. For example, Wang *et al.* [71] presented a power consumption anomaly detection method based on long short-term memory (LSTM) point forecasts and error pattern. In [72], Krishna *et al.* adopted Principal Component Analysis (PCA) and density-based spatial clustering on noise pattern to detect the anomalies which are deviations from the normal electricity consumption behavior. Kim *et al.* [73] presented a framework which utilizes spatial and temporal correlation between multiple solar farms to defend against data integrity attacks and learns the inter-farm/intra-farm correlation between measurements to perform anomaly detection. However, most of the existing anomaly detection applications for power systems are deterministic and thus insufficient to characterize the uncertainties of cyber attacks. Probabilistic approaches that provide quantitative uncertainty information associated with cyber attacks are therefore expected to better assist power system operations.

To address the aforementioned limitations, in this section, a data-driven probabilistic anomaly detection methodology is developed to provide reliable defense strategies against various cyber attack scenarios. First, a deep neural network, LSTM, is used to model the temporal dependencies within the LMP profile and correlations with explanatory variables. Then, a parametric probabilistic forecasting model is adopted to convert the LMP point forecasts to probabilistic forecasts, which is used for anomaly detection. Our major contribution is to formulate the anomaly detection problem as a probabilistic forecasting task and implement this approach to the publicly available electricity market data. The proposed probabilistic anomaly detection algorithm utilizes prediction interval to reveal the underlying structures within normal behavior and detect unexpected events.

The rest of the section is organized as follows. Subsection 4.1.2 describes the proposed probabilistic anomaly detection method, which consists of a deep-learning based deterministic forecasting model and a parametric probabilistic forecasting model. Subsection 4.1.3 applies and validates the developed probabilistic anomaly detection method to two types of cyber attack scenarios. Concluding remarks are discussed in Subsection 4.1.4.

4.1.2 Methodology

The overall framework of the proposed probabilistic anomaly detection methodology is illustrated in Fig. 4.1. It consists of three major steps:

1. Step 1 (gray blocks): Feed the historical data into an LSTM based forecasting machine to predict LMP.

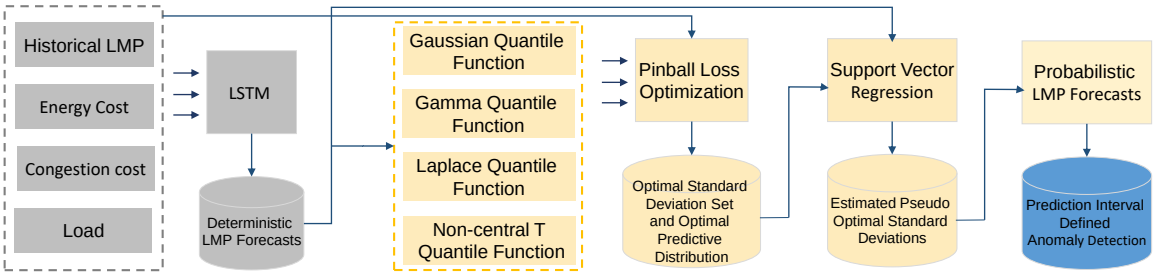


Figure 4.1: The Overall framework of the probabilistic anomaly detection model for electricity market data

2. Step 2 (orange blocks): Convert the point forecasts to prediction intervals (PIs) using a parametric probabilistic forecasting method based on designated predictive distribution shapes and pinball loss optimization.

3. Step 3 (blue block): Detect anomalies based on the threshold confidence and evaluate the performance.

The details are explained in the following content.

Multi-input Long Short Term Memory

Due to data availability, it is impractical to collect all explanatory variables (e.g., temperature and humidity, etc.) to build an ideal LMP forecasting model. In this section, we select the energy cost, congestion cost, forecast load, and their corresponding lagged variables to train the LSTM model, since they are published in real-time for most electricity market operators.

LSTM is a special recurrent neural network (RNN) architecture for time series modeling and forecasting, which has the capability of learning and memorizing long-term dependencies within the time-series data. The basic topology of standard RNN is shown in Fig. 4.2, where X denotes input, Y denotes output. h is the hidden state, W_{hx}, W_{yh}, and W_{hh} are the weight matrix among inputs, outputs, and hidden state itself, respectively. The standard RNN has one hidden layer, which could only trace back to a few time steps due to the vanishing gradient effect [74]. To better capture the long-term dependencies, LSTM introduces different gates which could regulate the gradient flow of the network. Following the work of [75], the inner structure of the LSTM unit is illustrated in Fig. 4.3 and described in Eq.4.1.

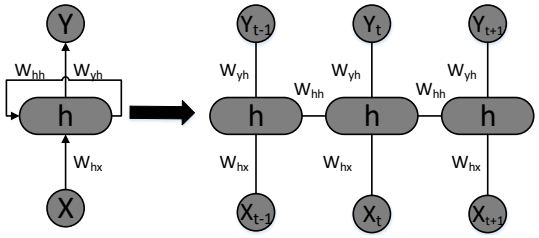


Figure 4.2: The structure of RNN

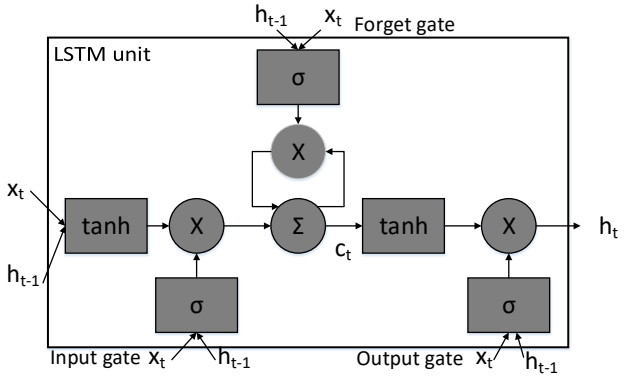


Figure 4.3: The inner structure of LSTM unit

$$
\begin{aligned}
i_t &= \sigma(x_t W_{ix} + h_{t-1} W_{ih} + c_{t-1} W_{ic} + b_i) \\
f_t &= \sigma(x_t W_{fx} + h_{t-1} W_{fh} + c_{t-1} W_{fc} + b_f) \\
c_t &= c_{t-1} f_t + i_t \cdot tanh(x_t W_{xc} + h_{t-1} W_{ch} + b_c) \\
o_t &= \sigma(x_t W_{ox} + h_{t-1} W_{oh} + b_o) \\
h_t &= o_t \cdot tanh(c_t)
\end{aligned}
\tag{4.1}
$$

where $i_{(\cdot)}$, $f_{(\cdot)}$, and $o_{(\cdot)}$ are the input gate, forget gate, and output gate, respectively. σ denotes the sigmoid activation function, h_t is the state at t, x_t denotes input, o_t is the cell output, and c_t is the memory state. LSTM updates its hidden state c_t by using the current input x_t and the previous state c_{t-1}. The final state h_t is determined by c_t and o_t. The weights are optimized by minimizing the difference between the LSTM outputs and training samples. In this study, the input vector of the multi-input LSTM can be expressed as:

$$x_t = [y_{t-1}, \Phi_t] \tag{4.2}$$

where Φ_t denotes the feature vector of the time step t, y denotes the observation at time step t.

Probabilistic Anomaly Detection

Once the deterministic LMP forecasts are generated, a multi-distribution database is formulated to model the possible shapes of the LMP predictive distribution. These four distributions, characterized by mean value (μ) and standard deviation (σ), are Gaussian, Gamma, Laplace and non-central-t distributions. The mean value is approximated by the deterministic point forecast and the standard deviation σ is calculated by minimizing the pinball loss of the quantile function at each time step. Based on the optimal pinball loss values, we select the best predictive distribution. The pinball loss value of a certain quantile L_m is expressed as:

$$L_{m,t}(q_{m,t}, y_t) = \begin{cases} (1 - \dfrac{m}{100}) \times (q_{m,t} - y_t), & y_t < q_{m,t} \\ \dfrac{m}{100} \times (y_t - q_{m,t}), & y_t \geq q_{m,t} \end{cases} \tag{4.3}$$

where y_t represents the tth observation, m represents a quantile percentage from 1 to 99, and q_m represents the predicted quantile. For a given m percentage, the quantile q_m represents the value of a random variable whose cumulative distribution function (CDF) is m percentage. Pinball loss is one of the most popular metrics for evaluating probabilistic forecasts [76]. Smaller pinball loss values indicate better probabilistic forecasting.

The process of probabilistic anomaly detection is described as follows:

1. Parameterizing the quantile in terms of μ and σ, where μ assumes to be the point forecast. The mth quantile of the tth point forecast, $q_{m,t}$ is expressed as:

$$q_{m,t} = F^{-1}(\frac{m}{100}, \hat{y}_t, \sigma_t) \tag{4.4}$$

   where, $\hat{y}_t$ and y_t are deterministic forecasts and observations, respectively. $F^{-1}(\cdot)$ is the inverse CDF function. The corresponding pinball loss is expressed as Eq. 4.3.

2. Calculating the unknown parameter σ at each time step by minimizing the averaged sum of pinball loss through genetic algorithm (GA) [77]:

$$\begin{aligned} \sigma_t^* &= \arg\min_{\sigma_t} \frac{1}{N_m} \sum_{m=1}^{N_m} L_{m,t}(\sigma_t, y_t, \hat{y}_t, m) \\ &\textit{subject to} \qquad \sigma_l \leq \sigma_t \leq \sigma_u \end{aligned} \tag{4.5}$$

   where σ_t^* is the optimal standard deviation of the tth time step; $N_m = 99$ is the number of quantiles; σ_l and σ_u are the lower and upper bound of σ, which are set as 0.01 and 80, respectively.

3. A support vector regression (SVR) surrogate model [78] is used to fit the point fore-

cast and σ^* in the training stage, which is used to generate unknown pseudo optimal standard deviations, $\hat{\sigma}^*$, in the forecasting stage.

4. During probabilistic forecasting, both the deterministic forecasts, i.e. μ, generated by LSMT and the estimated pseudo optimal standard deviation $\hat{\sigma}^*$ generated by SVR are used to determine the prediction interval (PI) [79].

5. For each time step, the observation falls into a certain PI, which is used to estimate the likelihood of it being anomaly (outliers). The deterministic prediction decides the best estimate of next step LMP, while the probabilistic prediction quantifies the uncertainty of all possible observations. The larger PI denotes further deviation from its nominal value. In this section, we assume an anomaly is spotted whenever the observation falls out of the 70% PI. This detection threshold can be further tuned through a sensitivity study which is out of our scope.

4.1.3 Case Study

Data Description

For data preparation, we first built an electricity market simulator based on Matpower [53] using a combined model of day-ahead economic dispatch (ED) and real-time incremental economic dispatch (IED). We then run the simulator on the IEEE 14 bus system with 11 loads selected from PJM load profiles. The day-ahead hourly load forecast and 5-minute real-time load forecast from Sept. 19th to Oct. 17th 2019 were used for ED and IED, separately. The simulated 5-minute real-time LMP, energy cost, congestion cost and the 5-min load forecast were used for training and testing. The ratio of the number of training samples to testing samples was 3:1. The LSTM deterministic forecasting model has two hidden layers of 50 and 30 neurons and the weights were optimized with Adam.

Cyber Attack Scenario Design

Two kinds of attacks are implemented in the simulator: Load Redistribution Attack (LRA) and Price Responsive Attack (PRA).

LRA was first introduced by Yuan et al [30], as a kind of FDIA attack where only the measurements related to some load bus power injection are attacked. LRA redistributes the load by increasing/decreasing certain loads at some buses while keeping the total load unchanged [65]. Since no attacks happened on the well-protected generation buses and LRA can bypass bad data detection, LRA can be hard to detect in real-time. The damage of LRA is that it can lead to a wrong dispatch result, i.e., fake solution from the economic

dispatch problem, which may overload certain transmission lines and raise LMPs. The LRA was added to the simulator before solving the IED problem to redistribute the 5-min load forecast. It tries to increase the load prediction in the targeted bus, while decreasing the load prediction in the non-targeted buses. To maximize the gain of the attacks, LRA is only activated in the simulator during the critical hours, i.e., when LMP has a big change in the historical data. In our case, we observed that LMP changes dramatically during 11:00 to 13:00, when LMP increases due to line congestion, and 20:00 to 22:00, when LMP decreases due to the removal of congestion. Therefore, we only add attacks during these two time periods, to extend the time of line congestion. LRA is added by the following procedure:

1. Within the time period 11:00 to 13:00, check if we have already applied attacks in the previous steps. If the number of existing attacks is greater than the maximum allowed attacks, terminate.

2. Check if the next total load prediction is greater than the current total load by 5%. If yes, we reduce the load increasing rate at the non-targeted bus where its corresponding load is increasing, and make the load decreasing rate higher at the non-targeted bus where its load is decreasing. We then apply the adjusted load to the targeted bus to increase its incremental load and accelerate the LMP ramping.

A similar procedure is applied to the time period 20:00 to 22:00, where we aim to slow down the load decreasing rate at the targeted bus, in order to extend the period of line congestion.

PRA is a type of LAA, inspired by the real-time pricing attacks [64], and Manipulation of Demand attack (MAD) [67], which change the load behaviors to damage the power grid. The motivation of our PRA is that the quick growth of smart grid foresees the wide usage of load management technologies, which can change the load behaviors based on the current LMP information. For example, the controller of a smart appliance can switch to the full-power mode when the price is low, and keep in energy saving mode when the price is high. Unlike the infrastructures of power grid, which are well-protected, the load controllers are located in the user end with much less security to defend against cyber attacks. The PRA is designed by injecting false price signal to the load controllers so as to inverse the controller logic, to use more power when LMP is high and there is a high opportunity of line congestion in the power grid. By increasing the load demand at such a critical time period, we expect it can possibly change the LMP by introducing more congestion.

Assuming there is a delay in the control of price-responsive demand after LMP changes, e.g., the load change happens 30 minutes after the price change. Our implementation of PRA is summarized as follows:

1. Check if we have already applied attacks in the previous steps. If the number of existing attacks is greater than the maximum allowed attacks in the given time period, terminate. (In our test case, we check if there are 5 continuous attacks happened during the last 5 time steps.)

2. Check if LMP has a big increase, compared to the LMP from last step. (In our test case, we checked if it had an increase over 10%.) If yes, adjust the price-responsive load by the following equation

$$PD_{adjusted} = PD_{base} + PD_{pr} * (\lambda_{curr}/\lambda_{pred})^{\beta} \tag{4.6}$$

where PD_{base} and PD_{pr} are the base load and the price-responsive part of the load, similar to the definition given in [64]. In our test case, we set $PD_{base} = 90\% * PD_{pred}$, i.e., 90% of the load forecast, and the rest of the load are the price-responsive load. The decreasing factor β is set to be -0.8. Note that our formulation is slightly different from the one used in [64], since we use day-ahead LMP prediction λ_{pred} as a base line. If the current LMP is equal to the predicted one, the above equation becomes $PD_{adjusted} = PD_{pred}$.

Deterministic LMP Forecasting Results

Three evaluation metrics are used to assess the deterministic forecasting accuracy, which are the normalized root mean squared error (nRMSE), normalized mean absolute error (nMAE), and mean absolute percentage error (MAPE). For these metrics, a smaller value indicates better forecasting performance. Deterministic LMP forecasting results are summarized in Table 4.1. In this study, the persistence method (PS) is adopted as the baseline since it is superior for a shorter forecast horizon [80]. Overall, the accuracies of the LSTM deterministic LMP forecasts are better than those of persistence forecasts under both cases with or without attack. It is mainly because the LMP data is highly temporally correlated, and the LSTM model outperforms in capturing long-term dependencies.

Table 4.1: 5-min ahead LMP forecasting performance

Model	Metric	Scenario		
		w/o attack	LRA	PRA
LSTM	NMAE(%)	1.07	5.20	5.96
	NRMSE(%)	1.35	6.10	6.83
	MAPE(%)	2.28	7.99	8.43
PS	NMAE(%)	3.80	6.40	6.66
	NRMSE(%)	6.50	7.86	8.42
	MAPE(%)	4.71	9.13	10.27

Probabilistic Anomaly Detection Results

This section evaluated the performance of the proposed probabilistic anomaly detection method. Laplace distribution was selected as the predictive distribution based on its minimal pinball loss in the training process. Therefore, the LSTM model with Laplace distribution (LSTM-Laplace) was chosen as the final anomaly detection model. The performance skill scores were calculated based on Table 4.2, where The true positive (TP) denotes the number of detected attacks; false negative (FN), i.e., type II error, denotes the number of missed detection of attacks; False positive (FP), i.e, false alarm or type I error, denotes the number of normal data treated as attacks; true negative (TN) denotes the number of normal data correctly identified. N_s is the total number of test samples. Among these indexes, the FP can cause false alarms, which may add redundant work to system operators, while the FN missed by the detection model may bring loss to market end users.

Table 4.2: Contingency table of attack detection

	Attack (Yes)	Attack (No)	Total
Detected (Yes)	TP (hit)	FP (miss)	TP+FP
Detected (No)	FN (miss)	TN (hit)	FN+TN
Total	TP+FN	FP+TN	N_s=TP+FP+FN+TN

Evaluation Metrics

We calculated the true positive rate (TPR), false positive rate (FPR), and F1 score of the anomaly detection results. The mathematical expressions of the three metrics are expressed as:

$$TPR = \frac{TP}{TP + FN} \tag{4.7}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.8}$$

$$F\text{-}1 = \frac{2TP}{2TP + FP + FN} \tag{4.9}$$

where the TPR measures the proportion of actual attacks that are correctly identified, the FPR measures the portion of normal data mistakenly categorized as attacks, and the F-1 score is the harmonic mean of the precision and recall. For the TPR and F-1 score metrics, values approaching 1.0 indicate better performance, while for FPR metric, a value closer to 0 indicates better performance.

To show the effectiveness of the proposed LSTM-Laplace model three baseline models were selected for comparison: LSTM model with Gaussian distribution (LSTM-Gaussian),

LSTM model with Gamma distribution (LSTM-Gamma), and quantile regression (QR). The reasons for choosing these baseline models are: (i) QR is a widely used non-parametric probabilistic method [81]. Since the proposed LSTM-Laplace model is a parametric method, the QR baseline allows us to explore the performance between parametric method and non-parametric method; (ii) the LSTM-Gaussian and LSTM-Gamma model allow us to explore the detection performance based on different predictive distribution types.

The evaluation metrics of different models are compared and summarized in Table 4.3. Overall, the proposed LSTM-Laplace anomaly detection method has a higher TPR, F-1 Score, and lower FPR compared with other anomaly detection methods, which shows the effectiveness of the proposed probabilistic anomaly detection algorithm. Note also that the models of LSTM-Gaussian, LSTM-Gamma, and LSTM-Laplace perform similarly and better than the QR method, which indicates that the optimization can help achieve better detection performance with different predictive distribution types in parametric methods. In addition, it is shown that the scores of LRA is better than that of PRA. It is mainly due to the larger LMP magnitude change under LRA and higher PRA attack frequency.

Table 4.3: Probabilistic Anomaly Detection Results

Method	Attack Scenario	Metrics		
		TPR	FPR	F-1 Score
LSTM-Laplace	LRA	**0.91**	**0.19**	**0.89**
	PRA	**0.86**	**0.23**	**0.86**
LSTM-Gaussian	LRA	0.89	0.24	0.87
	PRA	0.85	0.24	0.88
LSTM-Gamma	LRA	0.86	0.23	0.87
	PRA	0.85	0.24	0.86
QR	LRA	0.79	0.33	0.84
	PRA	0.71	0.35	0.65

Note: The best TPR, FPR, and F-1 score among different models are marked in boldface.

Results Analysis

To better visualize the probabilistic anomaly detection results, the PIs of selected time period under LRA and PRA are illustrated in Fig. 4.4 and Fig. 4.5, respectively. It is observed that for most of the no attack periods, the LMP reasonably lies within the PIs. When the observation in the attack period falls out of the 70% PI , it is defined as a truth positive detection. It is seen that the PRA frequency in Fig. 4.5 is higher than that of LRA in Fig. 4.4, and the magnitude change of LMP under LRA is higher than that under PRA. However, under both scenarios, the high detection accuracy shows the robustness of the proposed method. The width of the PI varies with the LMP variability. When the LMP fluctuates more frequently, the PI tends to be wider, and thereby the uncertainty under PRA is relatively higher.

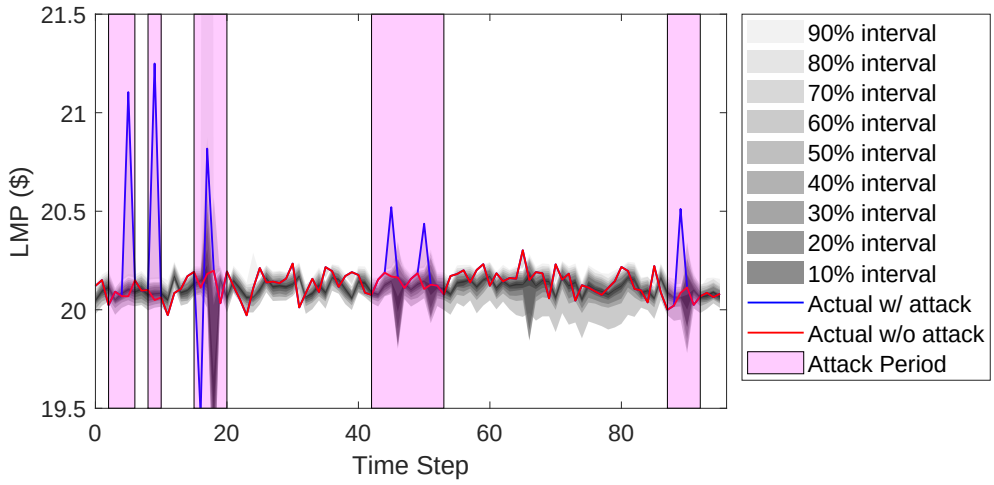


Figure 4.4: PIs of LMP under LRA attack

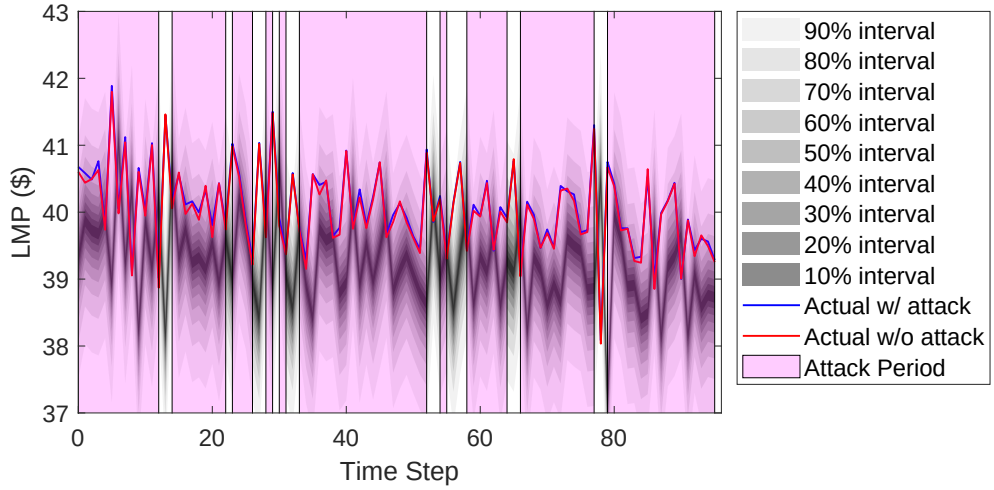


Figure 4.5: PIs of LMP under PRA attack

4.1.4 Conclusion

Results of the case study under different attack scenarios showed that the probabilistic anomaly detection method was able to effectively detect both LRA and PRA from electricity market data.

4.2 Real-time Point-wise Anomaly Detection - Part II Deterministic Methods

In this section, we explore the deterministic anomaly detection methods based on point-wise prediction. Unlike the probabilistic methods, deterministic methods will not predict the distribution of the next observation, rather, it produces a value closest to the true value based on historical data trends and other relevant features. The detailed implementation and testing are described below focusing on the IEEE 39-bus New England system.

4.2.1 Detection Algorithms and Detection Threshold

The deterministic detection starts with a one-step forward prediction on LMP data. It then compares the prediction with the true observation to learn the "normal" level of the system conditions. When under attack, the prediction error will surpass the predefined tolerance threshold since the attacked LMP data has never been observed in the historical trend. Hence, the deterministic anomaly detection algorithms contain two steps in general:

- Step 1: Predict the next step LMP and calculate the prediction error
- Step 2: Compare the prediction error with the detection threshold and report the anomalies

For Step 1, an accurate prediction algorithm can capture the correlations between the input features and the outcome predictions using both temporal connections and physical connections. The team started with the state-of-the-art deep learning algorithm, LSTM, and further compared with the transitional machine learning algorithms: Random Forests (RF), Gradient Boosting (GB), Support Vector Regression (SVR), Neural Network (NN) and Persistent Model (PS). Depending on the dynamics of the LMP data, each algorithm has its merits. Test results show that for the simulation dataset with piece-wise linear generation cost, the decision tree based algorithms (RF and GB) perform better since these LMPs are step-like signals. Meanwhile, the LSTM performs better on the PJM dataset and simulation data with quadratic generation cost function.

For Step 2, we consider a few parameters to calculate the best detection threshold. First is the confidence interval of the "normal" prediction defined by the quantile pair (upper and lower bound) of the errors. When a prediction error is outside of the quantile pair, we then consider the anomaly score (AS) defined below:

$$\text{AS} = P_{error} \cdot \exp\left(decay * state\right) \tag{4.10}$$

where P_{error}, representing the prediction error, is the difference of prediction and observation; *state* is an integer number that records the position of the current detection in its successive appearance after previous attacks. We keep track of *state* in half hour intervals which means

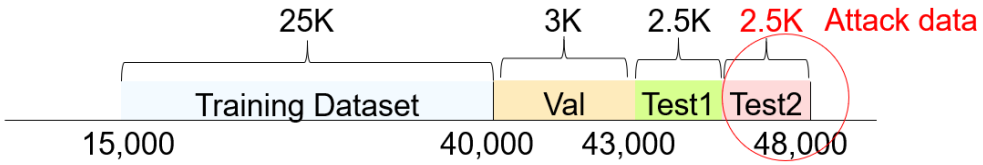


Figure 4.6: Data segmentation for deterministic anomaly detection algorithms.

for five minute LMP data, *state* ranges from 0 to 6. For example, *state* is 0 if P_{error} falls inside the quantile pair; it is 1 if P_{error} falls outside the quantile pair at the first time; it is 2 if P_{error} falls outside the quantile pair at the second time; and so on. *decay* is the decay factor that controls how fast the AS reduces with increasing state. We observed one single attack could produce long-term impact in the system which leads to high prediction errors even after attack is cleared. To reduce false alarms caused by such post-attack influence, we introduce a decay function defined as the exponential of the current attack state (i.e. *state*)regulated by a decay factor(i.e. *decay*), shown in Eq. 4.10. This formulation helps reduce the weight of the anomaly score when the attack is continuously following previous ones. *decay* can be selected from $[-0.1, -0.5, -1 - 10]$ covering full range of decaying speed.

The final detection result is determined by comparing the anomaly score (AS) with a selected threshold (short for AS thred). The threshold selection is carried out by searching through the three key parameters: quantile pair, *decay* and AS thred. Quantile pair defines the stringent level of initial filtering which creates the attack *state*. *decay* controls how fast AS decays and thus eliminates certain false alarms. AS thred is the final gate that decides the severity of the anomalies to be presented to the operators.

4.2.2 Evaluation Results

Test Setup

The team implemented the previously introduced three categories of attacks (FDIA, LRA and PRA) on the IEEE 39Bus system, a standard system extracted from New England Power Grid. The simulator is driven by the time series load data downloaded from PJM website. The details of the dataset generation were illustrated in Chapter 3. To implement the deterministic algorithms, we first segmented the data into four volumes. The first data chunk is used for training the prediction algorithms. The second chunk is for validation of the prediction. The third one is for testing and selecting the best quantile pairs and AS threshold. The last one, which is the only one that contains attacks, is used for final testing. This process is shown in Figure 4.6.

The input features for all algorithms are selected as the load features, LMP statistics, LMP laggings and time features, shown in Figure 4.7.

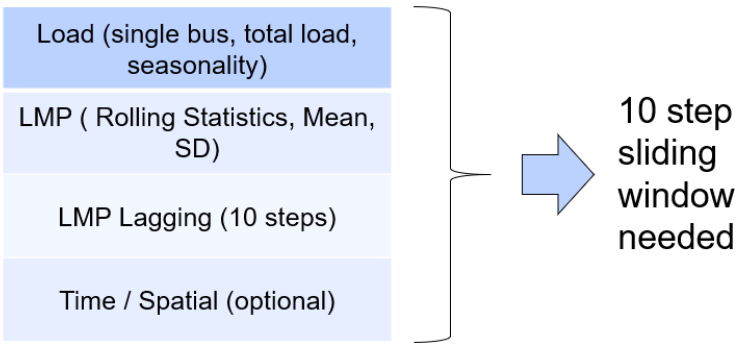


Figure 4.7: Features for deterministic anomaly detection algorithms.

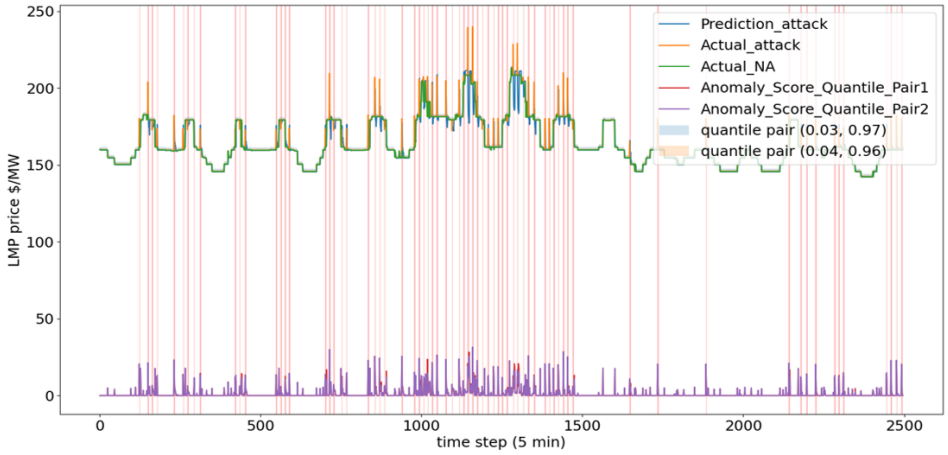


Figure 4.8: Time series plot of LMP and Anomaly Score at Bus 1 under FDIA.

Test Results on False Data Injection Attacks (FDIA)

The six algorithms are first tested on the FDIA. The total valid test data points are 2496 with 97 attack data points. The detector is applied on Bus 1 where we observe the LMP deviations under attacks (shown in Figure 4.8).

To better understand the detection results, we compared both prediction and detection performance of the six algorithms. The prediction performance is evaluated using root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), normalized root mean squared error (nRMSE), and normalized mean absolute error (nMAE). The key parameters for the prediction step are:

1. **LSTM** epochs = 100; batch_size = 128; drop_out=None; lr=0.0003
2. **RF** max_depth=10

Table 4.4: Prediction Performance Evaluation of Deterministic Methods on FDIA.

Algorithm	**RMSE**	**MAE**	**MAPE**	**nRMSE**	**nMAE**
LSTM	4.543493	1.682319	0.971247	0.018931	0.00701
RF	**4.045376**	**1.364551**	**0.787229**	**0.016856**	**0.005**
GB	4.401633	1.744481	1.00833	0.01834	0.007269
SVR	4.401106	1.418052	0.807473	0.018338	0.005909
NN	4.209041	1.811934	1.044549	0.017538	0.00755
PS	4.428706	1.202245	0.683726	0.018453	0.005009

Table 4.5: Detection Performance Evaluation of Deterministic Methods on FDIA.

Algorithm	**Decay**	**Quantile Pair**	**AS thred**	**DR**	**FAR**
LSTM	10	(0.03,0.97)	5.5	0.831	0.011
RF	**-1**	**(0.06,0.94)**	**6**	**0.859**	**0.006**
GB	10	(0.03,0.97)	6	0.803	0.012
SVR	10	(0.03,0.97)	6	0.845	0.011
NN	10	(0.03,0.97)	6	0.803	0.011
PS	10	(0.03,0.97)	6	0.831	0.014

3. **GB** random_state=0
4. **SVR** Kernal='rbf'; gamma=0.001; degree=5; epsilon=0.001
5. **NN** hidden_layer_sizes=(100,), activation='relu', solver='adam'

Performance results for the two cases are shown in Table 4.4 and Table 4.5, respectively.

The FDIA test results show the best performance algorithm is the RF with a detection rate (DR) of 85.9% and false alarm rate (FAR) of 0.6%.

Test Results on Load Redistribution Attacks (LRA)

LRA is applied only when system load is at its peak level and is greatly impacting system operation. This condition is very rare in normal operation, thus we only had 42 LRA attack data for testing. The Bus1 LMP and anomaly scores are shown in Figure 4.9.

Under the same parameters, the prediction and detection performance results are shown in Table 4.6 and Table 4.7, respectively.

For LRA attacks, the RF still keeps high prediction accuracy. However, the FAR of all algorithms increased significantly. This is because the LRA attacks cause much less impact on LMP than FDIA. This means to achieve a high detection rate, the detection threshold has to be decreased to identify the subtle changes in LMP. This can be observed from the low AS thred values in Table 4.7.

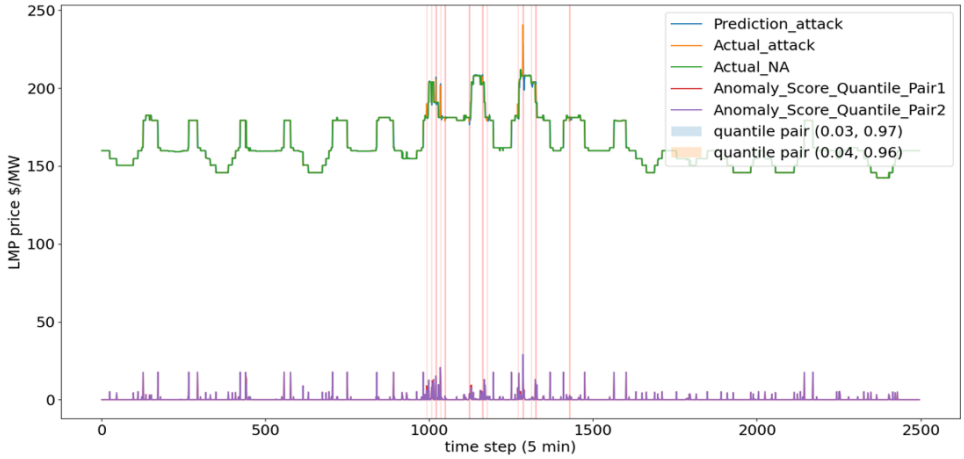


Figure 4.9: Time series plot of LMP and Anomaly Score at Bus 1 under LRA.

Table 4.6: Prediction Performance Evaluation of Deterministic Methods on LRA.

Algorithm	**RMSE**	**MAE**	**MAPE**	**nRMSE**	**nMAE**
LSTM	3.581589	1.206153	0.684184	0.014908	0.005021
RF	**3.099357**	**1.042749**	**0.60247**	**0.012901**	**0.00434**
GB	2.997857	1.047453	0.606068	0.012479	0.00436
SVR	3.816473	1.143128	0.638875	0.015886	0.004758
NN	3.701501	1.445421	0.832996	0.015407	0.006017
PS	3.22407	0.761086	0.432574	0.01342	0.003168

Table 4.7: Detection Performance Evaluation of Deterministic Methods on LRA.

Algorithm	**Decay**	**Quantile Pair**	**AS thred**	**DR**	**FAR**
LSTM	-1	(0.04, 0.96)	1.5	0.857	0.0599
RF	-10	(0.05, 0.95)	1	0.81	0.042
GB	**-0.5**	**(0.07, 0.93)**	**2.5**	**0.881**	**0.034**
SVR	-10	(0.03, 0.97)	1.5	0.833	0.047
NN	-10	(0.03, 0.97)	2.5	0.81	0.0456
PS	-1	(0.05, 0.95)	2.5	0.833	0.0387

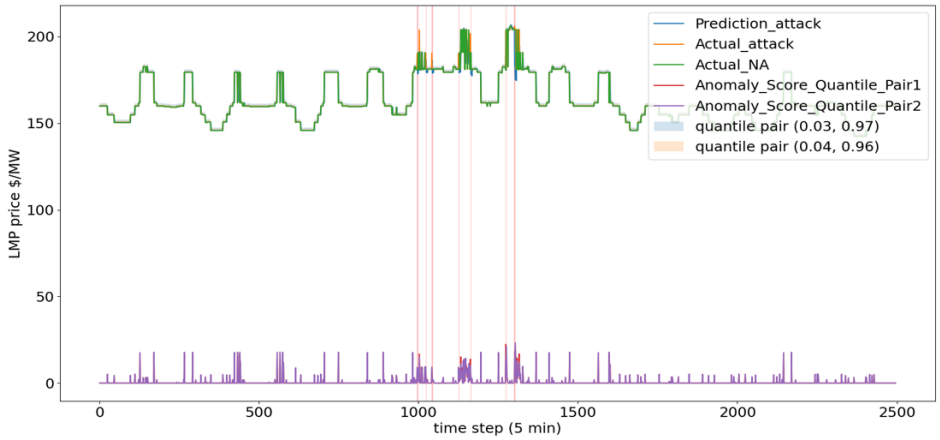


Figure 4.10: Time series plot of LMP and Anomaly Score at Bus 1 under PRA.

Table 4.8: Prediction Performance Evaluation of Deterministic Methods on PRA.

Algorithm	**RMSE**	**MAE**	**MAPE**	**nRMSE**	**nMAE**
LSTM	2.543518	0.645792	0.373003	0.012252	0.003111
RF	**2.234067**	**0.795726**	**0.466811**	**0.010762**	**0.003833**
GB	2.263343	0.909256	0.538206	0.010903	0.00438
SVR	2.632484	0.626418	0.360789	0.012681	0.003017
NN	2.511591	1.13822	0.670077	0.012099	0.005483
PS	2.678541	0.594861	0.343866	0.012903	0.002865

Test Results on Price Responsive Attacks (PRA)

Further, we tested the algorithms on PRA attacks. The applicable conditions for PRA attacks are more restricted. The successful implementation of a PRA attack depends on load dynamics, as it compromises the feedback loop of demand response controllers. The more incremental/decremental demands, the more severe the PRA attacks can be. Using the PJM load profile, we were only able to produce 7 PRA attacks, shown in Figure 4.10. Under the same parameters, the prediction and detection performance results are shown in Table 4.8 and Table 4.9, respectively.

Similar with the LRA attack, we observe a higher FAR for PRA. The reason is because the sample number is too small and the distribution is highly biased. Among 7 attacks, all algorithms can detect 6, but to achieve this, the FAR has to be increased. The persistent model shows a better performance in this test.

Table 4.9: Detection Performance Evaluation of Deterministic Methods on PRA.

Algorithm	Decay	Quantile Pair	AS thred	DR	FAR
LSTM	-0.1	(0.03, 0.97)	8	0.857	0.0237
RF	-0.1	(0.04, 0.96)	4.5	0.857	0.0377
GB	-0.1	(0.03, 0.97)	7.5	0.857	0.0188
SVR	-0.1	(0.04, 0.96)	6.5	0.857	0.0261
NN	-0.5	(0.03, 0.97)	2.5	0.857	0.0506
PS	**-0.5**	**(0.04, 0.96)**	**9**	**0.857**	**0.0176**

4.2.3 Conclusion

In this section, we tested the deterministic algorithms on the simulation dataset under FDIA, LRA and PRA attacks. The anomaly detection is designed to be a two-step process where step 1 is to predict the next time step LMP and step 2 is to compare the prediction error with a pre-defined threshold. We proposed a novel definition of the anomaly score to minimize the influence of single attack on multiple time steps. The test results on FDIA achieved the Milestone with best performance of 85.9% DR and 0.6% FAR. Testing on LRA and PRA are biased due to lack of valid attack data. However, under both attacks, we observe a trade-off between DR and FAR can be taken based on the requirement of system operations. All detectors in this section show ultra-low detection latency (<1ms) because the detection focuses on one node and the structure of the trained model is relatively simple (e.g. fewer neural network layers in LSTM or fewer trees in RF).

4.3 Real-time Point-wise Anomaly Detection - Part III Algorithm Ensemble

In the previous section, we observed the decision tree type of detectors perform better on LMP data using piece-wise linear generation cost. In this section, we extended the test on more of such detectors and generalized the performance evaluation to datasets with varying severity of FDIA attacks. An ensemble model is also developed to leverage the strengths of individual detectors. The details are elaborated in the following content.

4.3.1 Data Overview

We used the simulated FDIA dataset described in Section 4.2. In addition, we created two more datasets that contain more severe attacks. All detectors tested in this section used 3 months of training data and two weeks following for testing with the exception of dataset 3 which used 4 weeks of testing duration. The following figures show an example of Node 1 LMP from the 3 datasets with Figure 4.11 having the least severe FDIA out of the 3 sets,

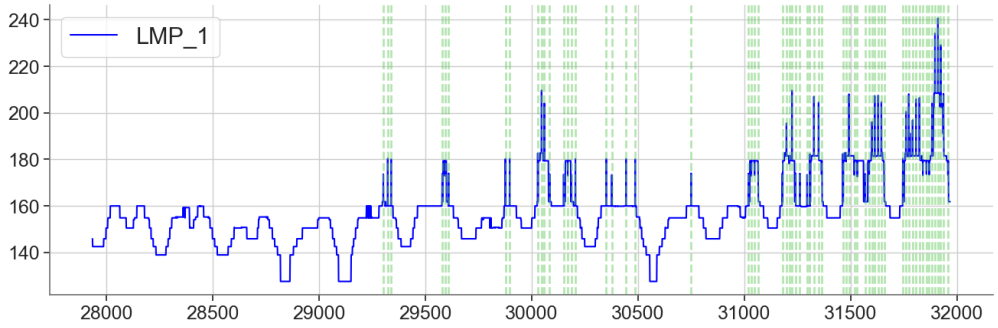


Figure 4.11: Dataset 1 example of LMP price at 1 node containing the least severe FDIA attacks with 63 attacks (green vertical lines) within a 2-week window.

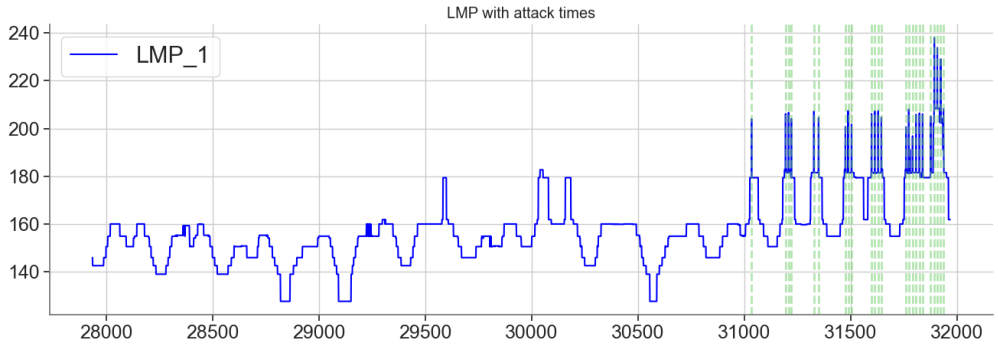


Figure 4.12: Dataset 2 example of LMP price at 1 node containing the average severity FDIA attacks with 24 attacks (green vertical lines) within a 2-week window.

Figure 4.12 average severity and Figure 4.13 the most severe. Within the 2 week testing window, dataset 1 has 63 attacks and dataset 2 has 24 attacks. Dataset 3 has 30 attacks in 4 week testing period.

4.3.2 Models & Hyperparameter Tuning

We first performed hyperparameter tuning for the four individual detectors: Random Cut Forest, Isolation Forest, K-Nearest Neighbor and Random Forest. To assess our models in a robust way we used K-fold cross-validation. Cross-validation is a statistical technique of evaluating model performance that is more stable than using a traditional splitting of a training set and a validation set. With each model we used 5 folds to achieve a robust accuracy performance, as shown in Figure 4.14.

In combination with k-fold cross validation, Scikit-Learn's RandomizedSearchCV [82] method was used, which allows us to define a grid of hyperparameter with a range of values, and randomly sample from the grid, while performing K-Fold cross validation with each

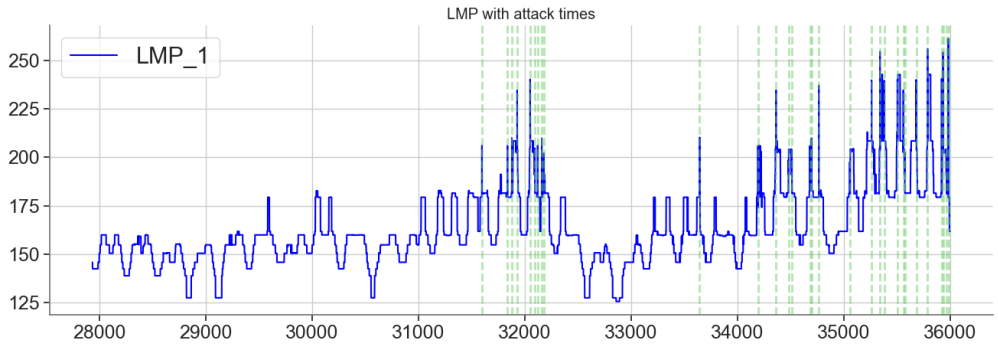


Figure 4.13: Dataset 3c example of LMP price at 1 node containing extreme severity FDIA attacks with 30 attacks (green vertical lines) within a 4-week window.

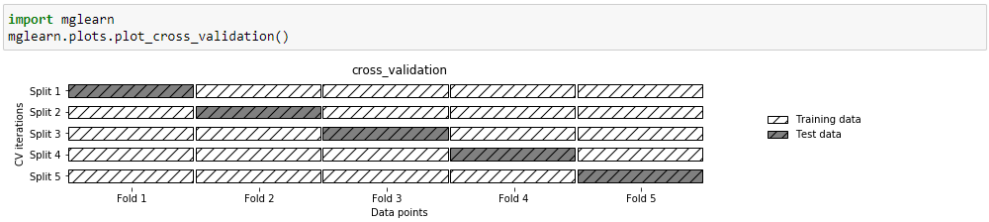


Figure 4.14: 5-fold cross-validation for assessing model performance.

combination of values. RandomizedSearchCV uses a method similar to Scikit-Learn's GridSearchCV which is an effective method for adjusting the hyperparameters of models and to improve the generalization performance of a model. With GridSearchCV, we try all possible combinations of the parameters of interest and find the best ones.

For each model we specify the hyperparameters we want to search and Scikit-Learn's RandomizedSearchCV performs all the necessary model fits. Traditional GridSearchCV is good when we work with a small number of hyperparameters. However, if the number of hyperparameters to consider is particularly high and the magnitudes of influence are unbalanced, the better choice is to use the RandomizedSearchCV which is what has been used with our models. In contrast to GridSearchCV, not all parameter values are tried out, but rather a fixed number of parameter settings is sampled from the specified distributions. The number of parameter settings that are tried is given by n_iter. n_iter parameter trades off runtime vs quality of the solution and in our case we used 10 iterations, totaling 50 fits given the 5 fold cross validation used. In the following content, we provide the results of hyperparameter tuning for each model.

Random Cut Forest

Random Cut Forest (RCF) [83] is an unsupervised algorithm for detecting anomalous data points which diverge from otherwise well-structured or patterned data. Anomalies can manifest as unexpected spikes in time series data, breaks in periodicity, or unclassifiable data

points. They are easy to describe in that, when viewed in a plot, they are often easily distinguishable from the "regular" data. Including these anomalies in a dataset can drastically increase the complexity of a machine learning task since the "regular" data can often be described with a simple model.

The main idea behind the RCF algorithm is to create a forest of trees where each tree is obtained using a partition of a sample from the training data. For example, a random sample of the input data is first selected. The random sample is then partitioned according to the number of trees in the forest. Each tree is given such a partition which organizes that subset of points into a k-d tree. The anomaly score assigned to a data point by the tree is defined as the expected change in complexity of the tree as a result of adding that point to the tree; which, in approximation, is inversely proportional to the resulting depth of the point in the tree. The random cut forest assigns an anomaly score by computing the average score from each constituent tree and scaling the result with respect to the sample size.

Hyperparameter Tuning:
The primary hyperparameters used to tune the RCF model are:

1) The number of trees indicated by *num_trees* parameter.
2) The number of samples per tree indicated by the *num_samples_per_tree* parameter.

Increasing *num_trees* has the effect of reducing the noise observed in anomaly scores since the final score is the average of the scores reported by each tree. During the random grid search cross validation:

1) The number of trees parameter values used: 10, 25, 50, 100, and 150.
2) The number of samples per tree parameter the values used: 64, 256, 512 and 1024.

The optimal parameters determined were:

1) The number of trees parameter 50 (default is 100).
2) The number of samples per tree parameter 512 (default is 256).

An example of the anomaly scores generated by RCF for a 2 week duration is shown in Figure 4.15.

Isolation Forest
Isolation forest [1] is an unsupervised learning algorithm for anomaly detection that works on the principle of isolating anomalies, instead of the most common techniques of profiling normal points. Isolation forest (IF), like any tree ensemble method, is built on the basis of decision trees. In these trees, partitions are created by first randomly selecting a feature and then selecting a random split value between the minimum and maximum value of the selected feature. In principle, outliers are less frequent than regular observations and are different from them in terms of values (they lie further away from the regular observations in the feature space). That is why by using such random partitioning they should be identified

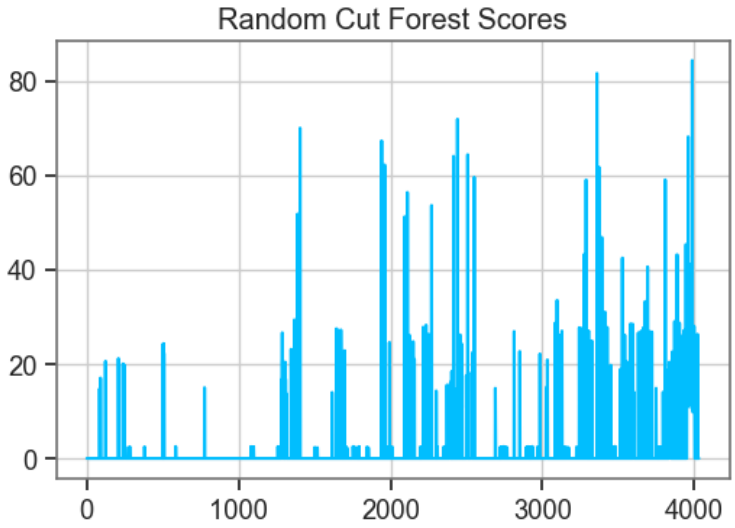


Figure 4.15: RCF anomaly score example for a 2 week duration.

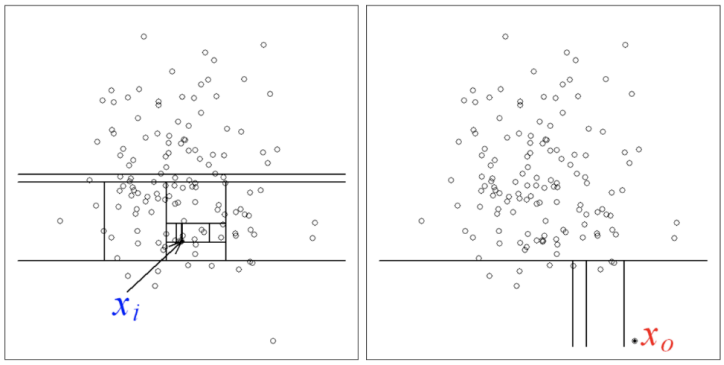


Figure 4.16: Identifying normal (more partitions) vs. abnormal observations (less partitions) [1].

closer to the root of the tree (shorter average path length, i.e., the number of edges an observation must pass in the tree going from the root to the terminal node), with fewer splits necessary.

The idea of identifying a normal vs. abnormal observation can be observed in Figure 4.16. A normal point (on the left) requires more partitions to be identified than an abnormal point (right).

Hyperparameter Tuning:
The primary hyperparameters used to tune the IF model are:

1) The number of base estimators in the ensemble indicated by parameter $n\_estimators$.

2) The number of max samples indicated by parameter *max_samples*.
3) The amount of contamination of the data set, i.e. the proportion of outliers in the data set. Used when fitting to define the threshold on the scores of the samples indicated by parameter contamination.
4) The number of features to draw from X to train each base estimator indicated by parameter *max_features*.
5) Whether the individual trees are fit on random subsets of the training data sampled with replacement or without replacement is performed indicated by parameter bootstrap.
6) Whether the number of jobs to run will be in parallel or not for both fit and predict indicated by parameter *n_jobs*.

During the random grid search cross validation:

1) The number of base estimator parameter values used: 50, 100, and 150.
2) The number of max samples parameter values used: 64, 256, 512 and 1024.
3) The amount of contamination of the data set parameter values used: 0.1, 0.15 and 0.2.
4) The number of features to draw from parameter values used: 1, 3 and 5.
5) Individual trees sampled with replacement or without replacement.
6) Run in parallel or not for both fit and predict.

The optimal parameters determined were:

1) The number of base estimator parameter values used: 100 (default is 100).
2) The number of max samples parameter values used: 256 (default is 'auto').
3) The amount of contamination of the data set parameter values used: 0.15 (default is 'auto').
4) The number of features to draw from parameter values used: 1 (default is 1).
5) Sampling without replacement. (default without replacement).
6) Not running parallel. (default not parallel).

An example of the anomaly scores generated by IF for a 2 week duration is shown in Figure 4.17 on the left. The anomaly score of an input sample is computed as the mean anomaly score of the trees in the forest. An example of the prediction of outliers generated by IF is shown in Figure 4.17 on the right. For each observation, IF detector tells whether or not (+1 or -1) it should be considered as an inlier according to the fitted model.

K-Nearest Neighbors (KNN)

K-nearest neighbors (kNN) [84] is a supervised machine learning algorithm that can be used

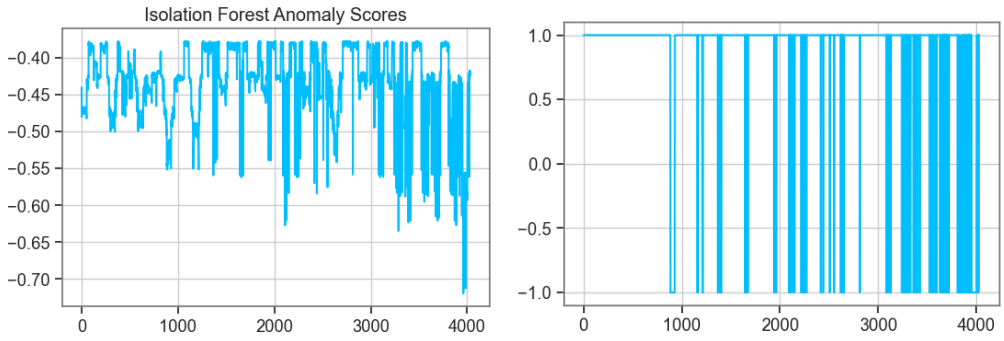


Figure 4.17: IF example for a 2 week duration of the anomaly score on the left and prediction score on the right.

to solve both classification and regression tasks. The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point and predict the label from these samples. The number of samples can be a user-defined constant which will be determined by computing the prediction errors. The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. The steps the model goes through in order to determine a class are as following:

1) Initialize the K value determined by the user.
2) Calculate the distance between test input and K trained nearest neighbors.
3) Check class categories of nearest neighbors and determine the type in which test input falls.
4) Classification will be done by taking the majority of votes.
5) Return the class category.

Hyperparameter Tuning:
The primary hyperparameters used to tune the KNN model are:

1) The number of neighbors indicated by $n\_neighbors$ parameter.
2) The weight function used in prediction indicated by the weights parameter. Most common are uniform and distance.
3) Algorithm used to compute the nearest neighbors indicated by the *weights* parameter.
4) Leaf size indicated by the $leaf\_size$ parameter.
5) The distance metric for the tree indicated by the *metric* parameter.

During the random grid search cross validation:

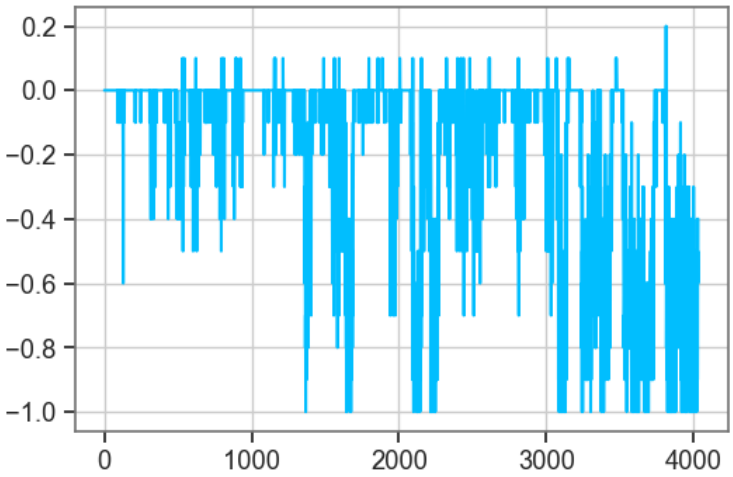


Figure 4.18: KNN distance score example for a 2 week duration.

1) The number of neighbors parameter values used was determined by computing the error rate of the model over a range between 1 and 40.
2) The weight functions used: uniform and weighted.
3) Algorithm used: ball_tree, kd_tree, brute and auto. Auto will attempt to decide the most appropriate algorithm based on the values passed to fit method.
4) Leaf size parameter values used: 10, 20, 30 and 40.
5) The distance metric used: Euclidean distance, Manhattan distance and Minkowski distance.

The optimal parameters determined were:

1) The number of neighbors parameter: 10 (default is 5).
2) The weight function parameter: uniform (default is uniform).
3) Algorithm parameter: auto (default is auto).
4) Leaf size parameter: 30 (default is 30).
5) The distance metric parameter: Minkowski distance (default is Minkowski distance).

An example of the distance scores generated by KNN for a 2 week duration is shown in Figure 4.18. The more negative value indicates more chance that the sample is considered to be an anomaly.

Radnom Forest (RF)

Random forest (RF), [85], like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits

out a class prediction and the class with the most votes is the final prediction.

Hyperparameter Tuning:

The primary hyperparameters used to tune the RF model are:

1) The number of trees in the forest indicated by parameter $n\_estimators$.
2) The number of features to consider when looking for the best split indicated by parameter $max\_features$.
3) The maximum depth of the tree indicated by parameter max_depth. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than $min\_samples\_split$ samples
4) The minimum number of samples required to split an internal node indicated by parameter $min\_samples\_split$.
5) The minimum number of samples required to be at a leaf node indicated by parameter $min\_samples\_leaf$. A split point at any depth will only be considered if it leaves at least $min\_samples\_leaf$ training samples in each of the left and right branches. This may have the effect of smoothing the model.
6) The boolean number indicating whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

During the random grid search cross validation:

1) The number of trees in the forest parameter values used: 50, 100, 150, 200, and 250.
2) The number of features to consider parameter values used: auto which is equal to number of features and sqrt which is the square root of the feature.
3) The maximum depth of the tree parameter values used: 5, 10, 15, 20, and None.
4) The minimum number of samples required to split parameter values used: 1, 2, 4, 5 and 6.
5) The minimum number of samples required to be at a leaf node parameter values used: 2, 3, 5 and 10.
6) Bootstrap samples are used when building trees or not.

The optimal parameters determined were:

1) The $n\_estimators$ parameter values used: 250 (default is 100).
2) The $max\_features$ parameter values used: sqrt (default is 'auto').
3) The max_depth parameter values used: 5 (default is 'None').
4) The $min\_samples\_split$ parameter values used: 5 (default is 2).
5) The $min\_samples\_leaf$ parameter values used: 2 (default is 1).

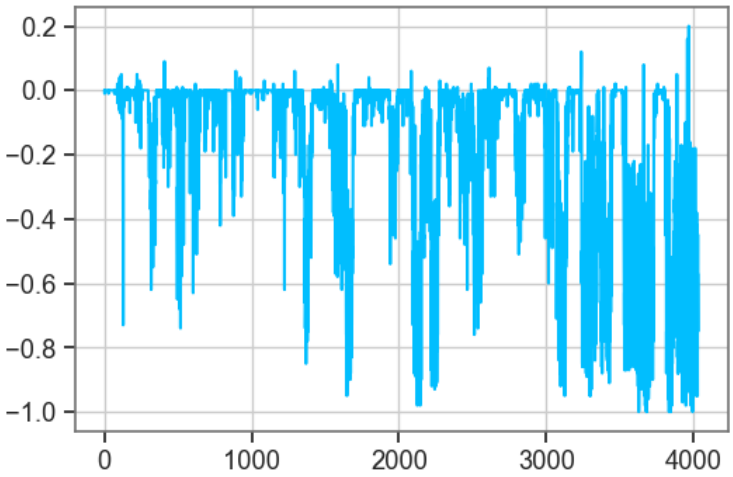


Figure 4.19: RF anomaly score example for a 2 week duration.

6) Bootstrap used when building trees. (default Bootstrap used).

An example of the scores generated by RF for a 2 week duration is shown in Figure 4.19. More negative values indicate higher levels of anomaly.

4.3.3 Threshold Optimization

Threshold optimization has been described in Section 4.2. With each model a few parameters were explored to find the optimal threshold needed to achieve the highest DR and lowest FAR. The parameters are:

1. Quantile pairs: (0.03 , 0.97), (0.04, 0.96), (0.05, 0.95), (0.06, 0.94), (0.07, 0.93)
2. Decay: (-0.1,-0.5,-1,-10)
3. AS thred: 0 to 5 with 0.1 increments and 5 to 30 with 0.5 increments.

Between all 3 parameters each model has 2000 combinations to search through in order to find the optimal threshold that maximizes the difference between DR and FAR. In Table 4.10, we listed the threshold parameters used to achieve optimal performance for each detector under each dataset.

4.3.4 Ensemble Method & Results

A voting ensemble or a "majority voting ensemble" is an ensemble machine learning model that combines the predictions from multiple models. It is a technique used to improve model performance, ideally achieving better performance than any single model. There are two

Algorithm	Dataset	Decay	Quantile Pair	AS thred	DR	FAR
RCF	Dataset 1	-0.1	(0.04, 0.96)	17	98.41	1.54
	Dataset 2	-0.1	(0.03 , 0.97)	12.5	100	3.02
	Dataset 3a	-0.1	(0.03 , 0.97)	26.5	100	0.07
	Dataset 3b	-0.1	(0.03 , 0.97)	28.5	100	0.05
	Dataset 3c	-0.1	(0.07, 0.93)	29.5	100	0.73
IF	Dataset 1	-0.5	(0.03 , 0.97)	1	98.41	1.94
	Dataset 2	-0.1	(0.03 , 0.97)	0.6	100	3.52
	Dataset 3a	-10	(0.03 , 0.97)	0.9	100	2.33
	Dataset 3b	-10	(0.03 , 0.97)	0.9	100	2.95
	Dataset 3c	1	(0.03 , 0.97)	0.7	93.33	9.29
KNN	Dataset 1	-0.1	(0.03 , 0.97)	0.7	100	2.32
	Dataset 2	-0.1	(0.03 , 0.97)	0.7	100	0.72
	Dataset 3a	-0.1	(0.03 , 0.97)	0.8	100	0
	Dataset 3b	-0.1	(0.03 , 0.97)	0.8	100	0
	Dataset 3c	-0.1	(0.03 , 0.97)	0.6	100	0.16
RF	Dataset 1	-0.1	(0.03 , 0.97)	0.7	100	0
	Dataset 2	-0.1	(0.03 , 0.97)	0.7	100	0
	Dataset 3a	-0.1	(0.03 , 0.97)	0.4	100	0
	Dataset 3b	-0.1	(0.03 , 0.97)	0.4	100	0
	Dataset 3c	-0.1	(0.05, 0.95)	0.5	100	0

Table 4.10: Threshold parameters selected for optimal performance.

approaches to the majority vote prediction for classification: hard voting and soft voting. We used hard voting for anomaly detection where the ensemble machine counts the number of predictions for each label and the label with the majority vote is the final result.

A voting ensemble may be considered a meta-model, a model of models. As a meta-model, it could be used with any collection of existing trained machine learning models and the existing models do not need to be aware that they are being used in the ensemble. In this section, we used the ensemble model on the aforementioned four detectors.

The model performance on all 3 datasets (low, medium, and high severity of FDIA) are shown in Table 4.11. The performance results reported are after optimizing the models' hyperparameters using randomized gride search with k-fold cross validations and optimizing the thresholds. The ensemble model results are also included in the table showing a performance of DR = 100% and a FAR $< 0.03\%$ with a processing delay time of < 27.5 millisecond.

	Data Set 1			Data Set 2			Data Set 3		
	DR	FAR	Delay	DR	FAR	Delay	DR	FAR	Delay
RCF	98.41%	1.54%	27.9	91.67%	1.40%	28.1	100%	0.73%	27.1
IF	98.41%	1.94%	0.03	100%	3.52%	0.03	93.33%	9.29%	0.03
KNN	100%	2.32%	0.35	100%	0.72%	0.31	100%	0.16%	0.35
RF	100%	0.00%	0	100%	0.75%	0	100%	0.16%	0
Ensemble	**100%**	**0.03%**	**28.3**	**100%**	**0.02%**	**28.4**	**100%**	**0.00%**	**27.5**
Confusion Matrix	3968	1	-	4007	1	-	8034	0	-
	0	63		0	24		0	30	

Table 4.11: Ensemble and model performance with the 3 datasets.

4.3.5 Conclusion

In this section, we presented the hyperparameter tuning and threshold selection for four individual anomaly detectors: RCF, IF, KNN, RF. We then applied the majority voting ensemble on the detection results to improve performance. The final results tested on three datasets show we have achieved the project milestone with $> 98\%$ detection accuracy, $<0.1\%$ false alarm rate and < 50ms computing delay.

4.4 Real-time Locational Anomaly Detection - Part I PJM Dataset

The goal of the point-wise anomaly detection is to identify potential attacks from the time series data flow. The detection results only indicate when an attack happens but do not answer where the attack happens. To further localize the attack region, we leverage the LMP signals at multiple locations to learn from their joint spatio-temporal correlations. This procedure generally includes two steps: (1) LMP signals are first clustered into smaller groups with similar dynamic behavior; (2) the nodes that deviate from the "group behavior" are then identified to describe the potential attack region. Since these nodes are the most sensitive nodes affected by the cyber-attack events, they indicate a neighbor region with highest probability to be the actual attack targets. We experimented with the locational anomaly detection on both PJM dataset and simulation dataset, which are elaborated separately in Section 4.4 and the subsequent Section 4.5.

4.4.1 Data Overview

The PJM dataset used in this section contains five-minute interval data of real-time market from September 1, 2019 to October 1, 2019. There are in total 8640 records with 2810 unique node IDs (called pnode ID in PJM database) from five transmission zones:

	LMP	Energy	Congestion	Marginal Loss
Mean	23.70	27.14	-3.11	-3.19
Std	58.61	54.38	27.94	2.64
Min	-580.11	4.44	-3.97	-317.22
Max	3,442.86	2,361.2	1,656.72	4,367.60

Table 4.12: Statistics of the PJM dataset.

PECO, BGE, DPL, COMED, and EKPC. The basic statistics of the numeric data (total LMP, energy cost, congestion cost and marginal loss) are shown in Table 4.12.

4.4.2 Clustering

We first clustered nodes into groups that exhibit similar behavioral patterns that a model can well represent. Two clustering methods have been explored: Hierarchical and K-means clustering. One of the evident disadvantages of hierarchical clustering is its high time complexity. Generally it is in the order of O(n* 2* log(n)), n being the number of data points. For K-means clustering, we can optimize some objective functions, e.g. within cluster Sum-Square, whereas in hierarchical clustering we do not have any actual objective function. K-means is determined to be the choice since hierarchical uses a lot of resources and is not suitable with large sets of data.

K-means is an unsupervised learning algorithm. The user chooses the "K" number of centroids used to define clusters. A point is considered to be in a particular cluster if it is closer to that cluster's centroid than any other centroid. It finds the best centroids by alternating between (1) assigning data points to clusters based on the current centroids and (2) choosing centroids based on the current assignment of data points to clusters. This process is carried out until all clusters remain unchanged.

Clustering Metrics

Two distance metrics were explored for clustering:

- Pearson Correlation
- Dynamic Time Warping (DTW): Designed to compute distance between 2 temporal sequences

The clustering performance of Pearson correlation and DTW were very similar. The clustered groups of all the 2810 nodes for one month duration matched by 97.2% with the transmission zone assignment. Correlation distance was calculated much faster than DTW and requires less computational resources. DTW has a quadratic time and space complexity that limits its use to only small data sets. FastDTW was then explored as an approximation of DTW that has a linear time and space complexity. Pearson correlation was chosen since it's still faster than FastDTW.

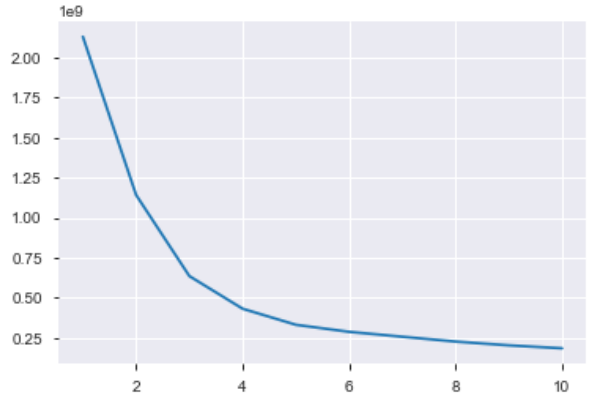


Figure 4.20: Elbow method using WSS vs Cluster number to optimize k.

K-means Optimization

A fundamental step for K-means is to determine the optimal number of clusters into which the data may be clustered. The Elbow Method is one of the most popular methods to determine this optimal value of k. The Elbow method looks at the total within-cluster sum of square (WSS) as a function of the number of clusters. Number of clusters are chosen so that adding another cluster does not improve much on the total WSS.

One disadvantage of the K-means algorithm is that it is sensitive to the initialization of the centroids. If a centroid is initialized to be a far point, it might just end up with no points associated with it, and at the same time, more than one cluster might end up linked with a single centroid. K-means starts with allocating cluster centers randomly and then looks for "better" solutions. K-means++ starts with allocating one cluster center randomly and then searches for other centers given the first one. K-means++ provides more speed and accuracy. This algorithm ensures a smarter initialization of the centroids and improves the quality of the clustering. Apart from initialization, the rest of the algorithm is the same as the standard K-means algorithm.

The Elbow method was applied to the 2810 nodes' LMP for a duration of a month and the results are shown in Figure 4.20. From the plot, a k of 5 was chosen. The assigned clusters contained: cluster 0: 457 nodes, cluster 1: 2103, cluster 2: 50, cluster 3: 169 and cluster 4: 31 as shown in Figure 4.21.

LMP and Congestion costs were both explored as features for clustering. In Figure 4.22, we see 100 node's LMP and congestion correlations. Results of the elbow method and centroid centers of both features were plotted in Figure 4.23.

For LMP and congestion cost, a k of 2, 3 and 4 were tested. After examining the clustering results, it was shown that LMP performed better in separating similar behavior nodes than congestion cost.

Another aspect that was explored with the correlation is the duration used to determine

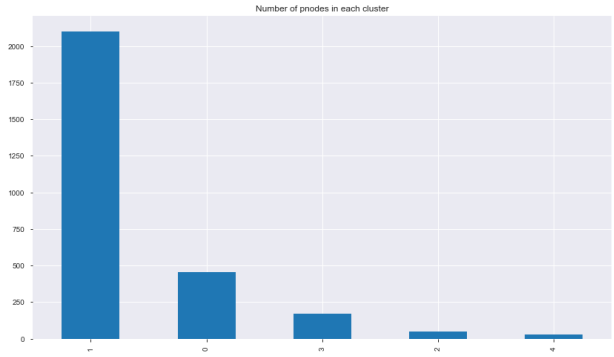


Figure 4.21: Histogram of cluster assignment.

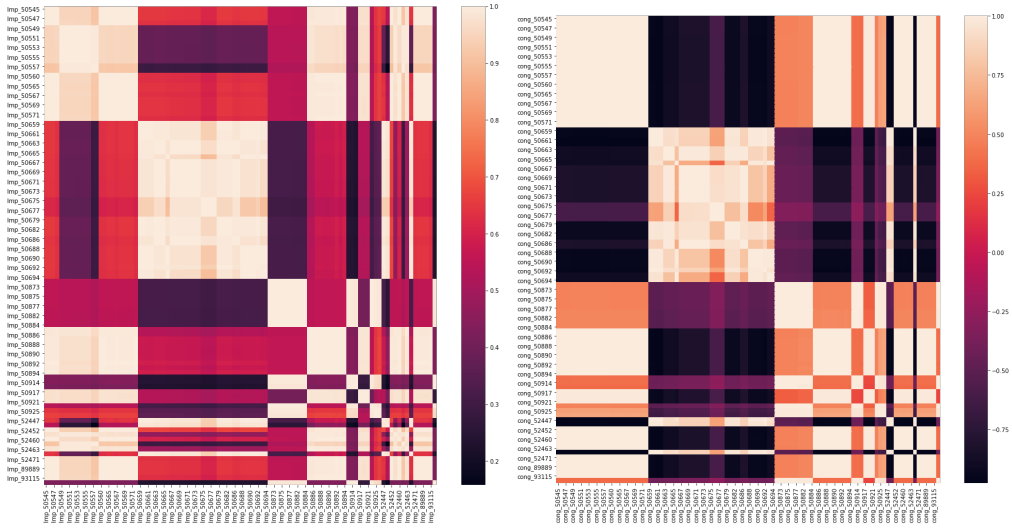


Figure 4.22: Correlation of LMP (left) and Congestion cost (right).

the clustering assignment. In Figure 4.24, we present three correlation plots. From the left to right each one represents 10 days in time with the left being the beginning of the month and the right being the end of the month. This shows how the correlation varies across time and therefore the cluster assignment.

In order to get a better understanding of this correlation variation over time, we compared 9 nodes to 1 reference node and computed the correlation hourly for a day (Figure 4.25). The plot shows that the node's correlations vary in time but they vary in a similar way.

We also looked at the distances from the nodes to the centroid center over time. Two examples are shown in Figure 4.26 and 4.27 for a duration of 10 hours each. We observed that even though the distance to the center varied over time, the variation happened to both nodes. Also, the assignment did not change often and when it did both nodes were reassigned together to a new cluster.

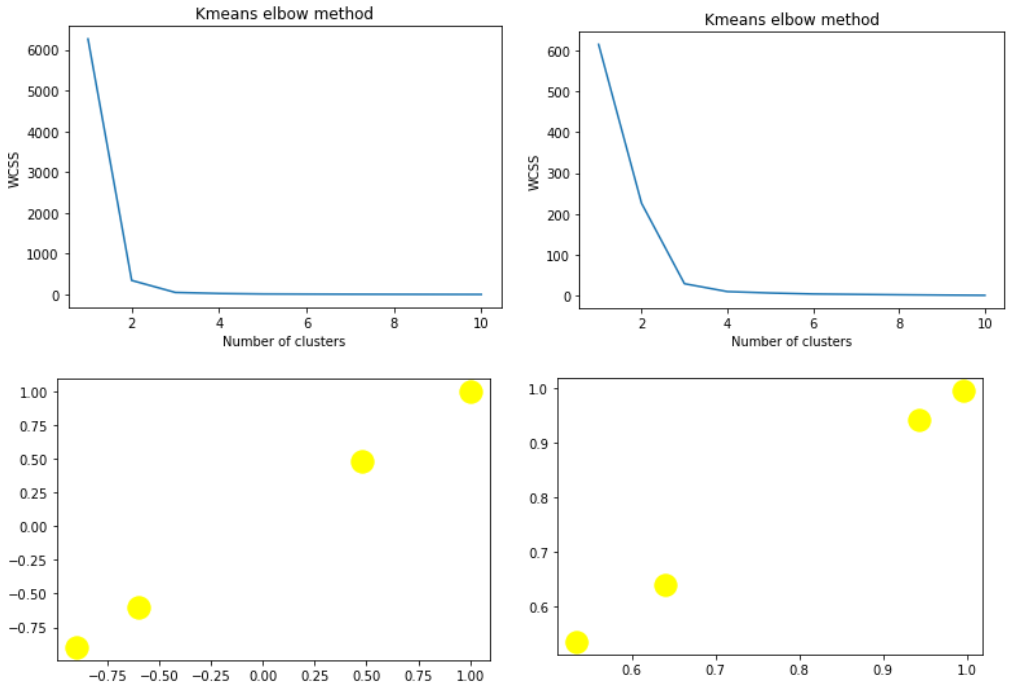


Figure 4.23: Elbow method and cluster centroids for LMP (left) and Congestion cost (right).

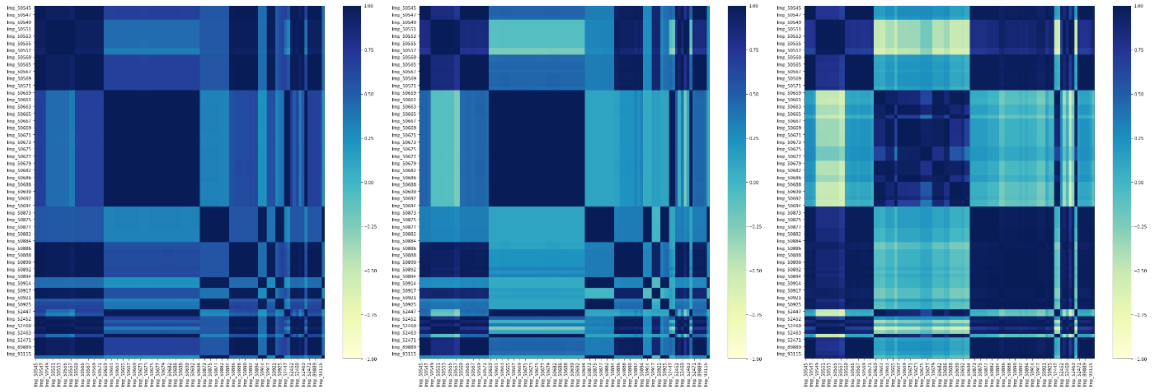


Figure 4.24: correlation over a period of 30 days divided into 3 correlations with 10 days each.

During the model training stage, we used 100 nodes from each of the 5 zones (i.e. Figure 4.28) as well as from the 5 clusters generated from the K-means process. In both cases we were able to get highly correlated nodes together and the model's performances were similar. Based on this fact, we measured the performance of clustering, by assuming that the 5 zones can represent the 5 clusters selected from the elbow method. The overlap between the zone

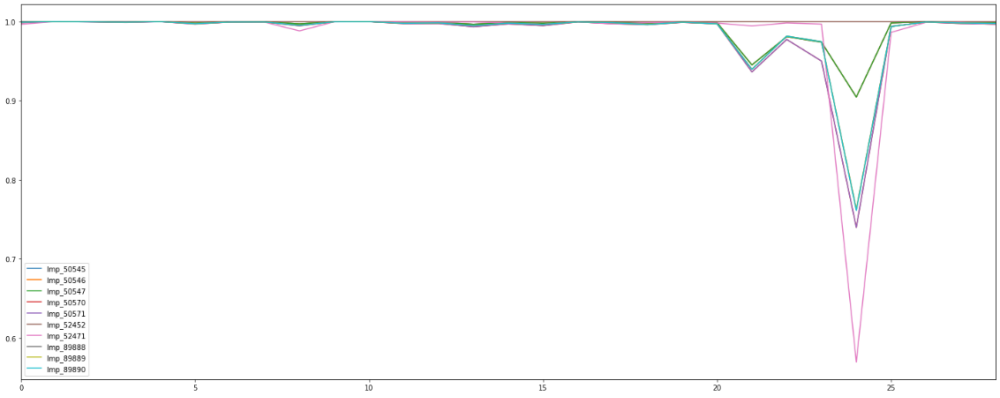


Figure 4.25: Correlation over time with 10 nodes.

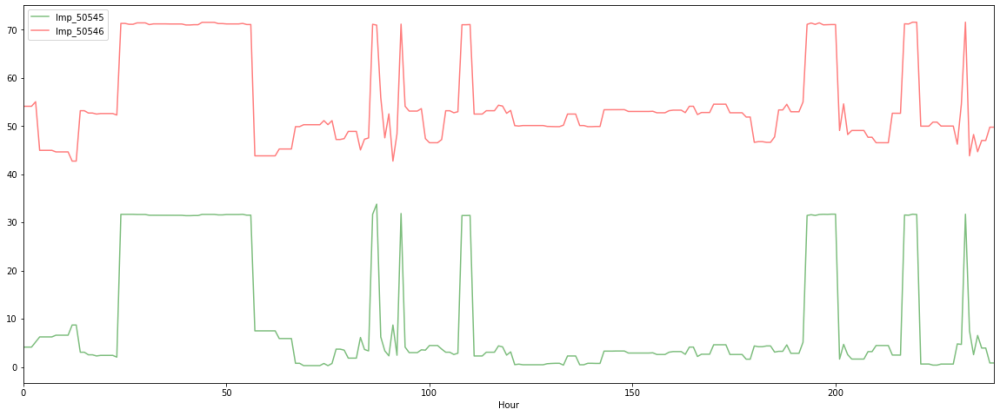


Figure 4.26: Distance between nodes 50545 and 50546 to the cluster center.

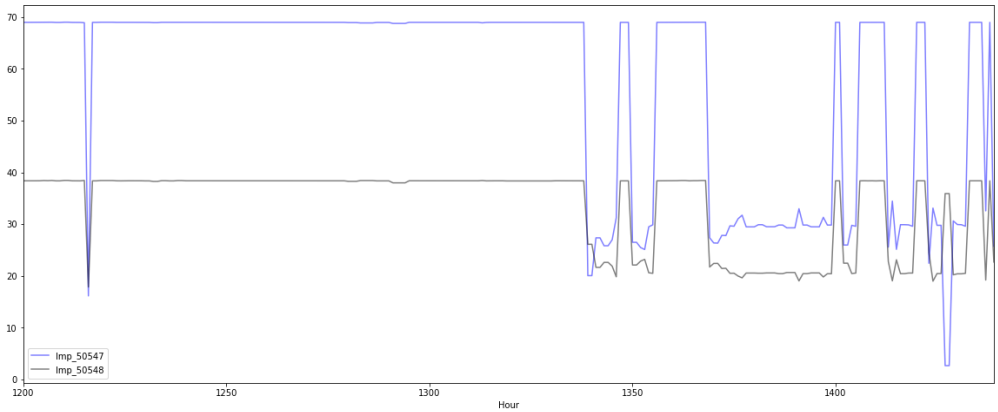


Figure 4.27: Distance between nodes 50547 and 50548 to the cluster center.

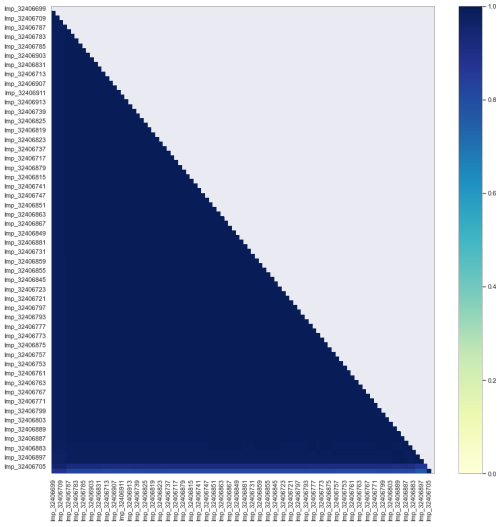


Figure 4.28: Example of 100 node's correlation from the COMED zone.

assignment and clustering assignment was 97.6% matched.

4.4.3 LSTM-Autoencoder

The LSTM network, introduced in Section 4.1, can be organized into an architecture called the Encoder-Decoder LSTM that allows the model to both support variable length input sequences and to predict or output variable length output sequences. In this architecture, an encoder LSTM model reads the input sequence step-by-step. After reading in the entire input sequence, the hidden state or output of this model represents an internal learned representation of the entire input sequence as a fixed-length vector. This vector is then provided as an input to the decoder model that interprets it at each step until the output sequence is generated.

We first clustered the raw data into small groups. The data used for training was from a cluster while testing data was a mixture of data from the same cluster and from other clusters. The idea is to see if the model is able to learn the behavior of the cluster and detect anomalies from testing data outside the cluster. Training data used was 20 days, validation data used was 5 days and testing data was 5 days (Figure 4.29). A preliminary set of hyperparameters was used: adam optimizer with MAE loss function, Relu activation function, batch size was 128 and epochs was 50.

Comparing the reconstruction errors with a threshold of 1.1 for the testing set within the cluster, we find 3 anomalies (Figure 4.30). On the other hand, with the same threshold for the testing set outside the cluster we detect 9 anomalies (Figure 4.31). This indicates that the model is able to learn the LMP patterns within its cluster. Note that the detected anomalies only appear at time points when the LMPs show different cluster behavior, i.e.

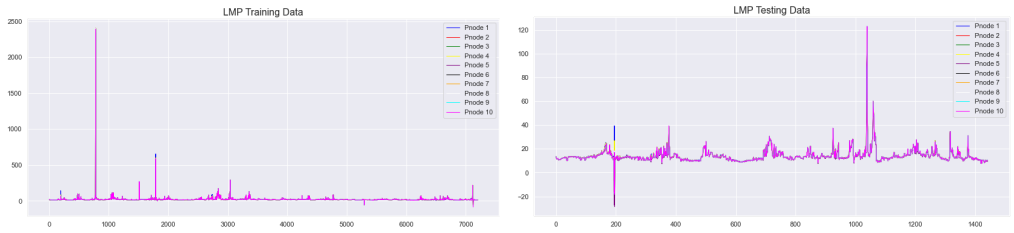


Figure 4.29: Training data set (left) and testing data set (right) from same cluster.

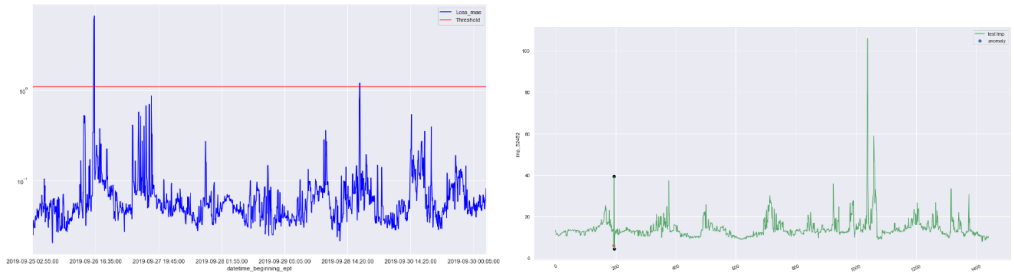


Figure 4.30: Reconstruction errors with threshold (left) and testing data set from same cluster as training data (right) with anomalies.

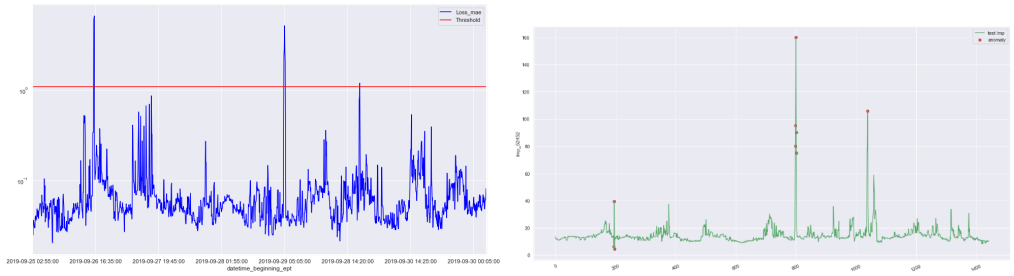


Figure 4.31: Reconstruction errors with threshold (left) and testing data set from outside cluster (right) with anomalies.

when new congestion patterns occur.

In order to extend this test to more nodes, a bootstrapping method was used. The idea is to use 100 nodes from a cluster and during training input 10 nodes at a time with replacement. This was done 500 times in order to explore all the data while keeping the inputs for the model set to 10 nodes.

Comparing the reconstruction errors with a threshold of 1.1 for the testing set within the cluster, we find 5 anomalies (Figure 4.32). On the other hand, with the same threshold for the testing set outside the cluster we detect 100 anomalies (Figure 4.33). This indicates that the model is able to learn the LMP patterns within its cluster using the bootstrapping method.

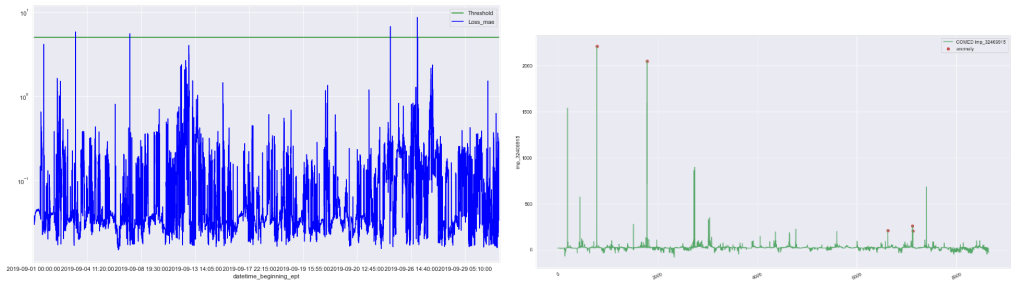


Figure 4.32: Reconstruction errors with threshold (left) and testing data set from same cluster as training data (right) with anomalies using bootstrap.

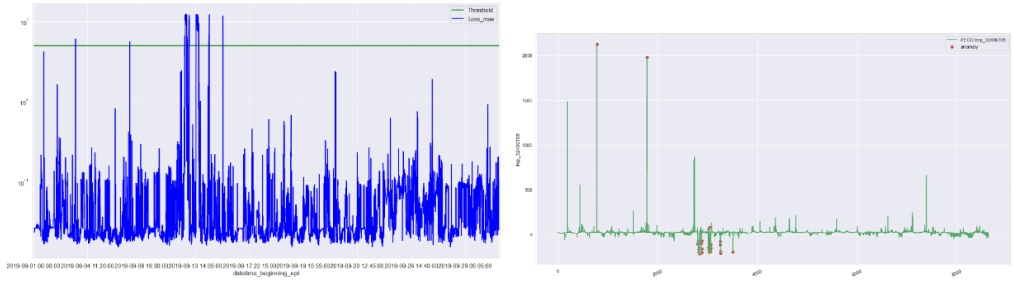


Figure 4.33: Reconstruction errors with threshold (left) and testing data set from outside cluster (right) with anomalies using bootstrap.

The bootstrap enhanced LSTM-Autoencoder was further tested on data from 5 zones. We trained 5 models representing each zone. For each model, the training set was from the zone itself, and the testing data set was from the zone and the 4 other zones outside the cluster. The expected outcome is that the number of anomalies from the test set that comes from the same zone as the training set to be less than all the other test sets.

4.4.4 LSTM-Autoencoder Optimization

After preliminary testing to prove the concept of cluster-based locational detection, we invested more effort to improve the performance and reduce the false detection. In the following content, we present the hyperparameter tuning for the LSTM-Autoencoder model.

Bottleneck Layer

In order to find the optimal number of neurons in the bottleneck layer, we tested the bottleneck layer from 1-10 and measured the model's training performance, defined as the number of anomalies detected. The optimal number of neurons for the bottleneck layer was 5 based on the searching results in Table 4.13. The same tuning process was further performed on all the nodes from each trained cluster, with results shown in Table 4.14. The best bottleneck layer should be able to find minimum anomalies in its own cluster but detect maximum

	Zero	One	Two	Three
Zero Model – [10,1,1,10]	100	2	3	8
Zero Model – [10,2,2,10]	28770	28800	22981	5760
Zero Model – [10,3,3,10]	8	1	9	9
Zero Model – [10,4,4,10]	23	2	6	7
Zero Model – [10,5,5,10]	0	3	7	6
Zero Model – [10,6,6,10]	10	5	10	8
Zero Model – [10,7,7,10]	6	2	3	14
Zero Model – [10,8,8,10]	1	0	3	10
Zero Model – [10,9,9,10]	0	2	11	7

Table 4.13: Number of anomalies detected by different bottleneck values – 100 nodes per cluster.

	Zero	One	Two	Three
Zero Model – [10,1,1,10]	19134	19249	19249	19296
Zero Model – [10,2,2,10]	13	43	37	63
Zero Model – [10,3,3,10]	1817	1453	1663	1381
Zero Model – [10,4,4,10]	8	0	0	16
Zero Model – [10,5,5,10]	76	124	71	229
Zero Model – [10,6,6,10]	9	1	0	13
Zero Model – [10,7,7,10]	17	0	1	9
Zero Model – [10,8,8,10]	14	1	0	8
Zero Model – [10,9,9,10]	39	0	0	12

Table 4.14: Number of anomalies detected by different bottleneck values – all nodes within cluster.

anomalies outside of the cluster.

Batch size

Batch size is the number of data points used to train a model in each iteration. Typical batch sizes are 32, 64, 128, and 256. Choosing the right batch size is important to ensure convergence of the cost function and parameter values, and to the generalization of our model. Batch size determines the frequency of updates. Using a bottleneck value of 5, batch values were varied from 32, 64, 128, 256 up to 512. Results in Table 4.15 did not show significant improvement of model performance with varying batch size.

Concurrent optimization

We then considered concurrent searching of multiple parameters to cover all possible combinations. Bottleneck layer varied between 1 and 5, batch size among 32, 64, and 128, training data size ranging from 1 day, 2 days, 1 week to 2 weeks, and time steps among 1, 3 and 5 in

	Zero	One	Two	Three
Zero Model – Batch Size = 32	8	2	1	11
Zero Model – Batch Size = 64	36	25	81	23
Zero Model – Batch Size = 128	19	8	31	22
Zero Model – Batch Size = 256	46	5	1	40
Zero Model – Batch Size = 512	9	1	0	11

Table 4.15: Number of anomalies detected by different batch sizes – all nodes within cluster.

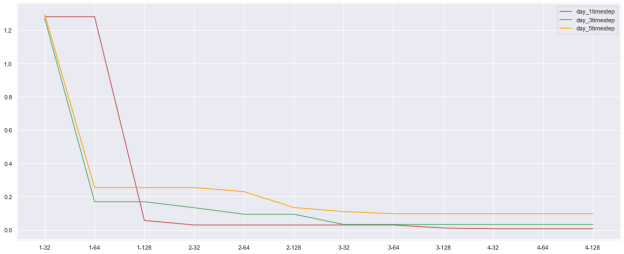


Figure 4.34: Losses minimum while varying bottleneck layer, batch size and time steps.

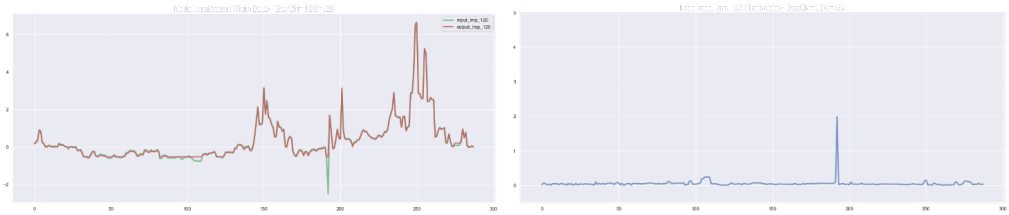


Figure 4.35: Input/ Output from model with training data (left) and its reconstruction errors (right).

order to form 36 different combinations. The optimal number of neurons for the bottleneck layer was determined to be 4, the batch size 128 and time step of 1. Figure 4.34 shows the minimum loss during training for 1 day of data, on the x-axis the first number indicates the bottleneck number (1-4) and the second number indicates the batch size (64-128). This allowed us to optimize bottleneck layer, batch size and time steps simultaneously.

Results from the model using the optimal hyperparameters with the training data and its reconstruction errors are presented in Figure 4.35.

Testing the same model with a different node shows similar performance as the training data Figure 4.36. Both examples in Figure 4.35 and 4.36 indicate high errors when reconstructing negative LMP values since they are rarely seen in the training data.

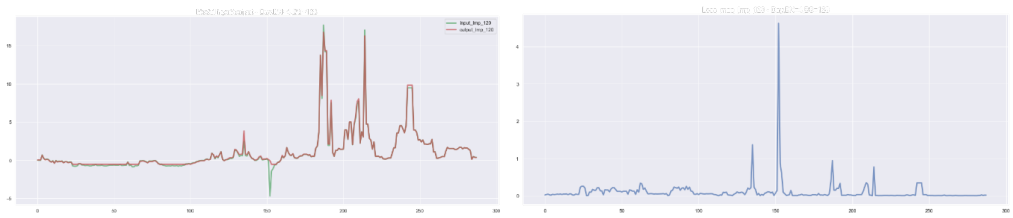


Figure 4.36: Input/ Output from model with testing data (left) and its reconstruction errors (right).

4.4.5 Conclusion

In this section, we presented the concept of a two-step locational detection framework and its implementation on PJM dataset. The first step is to cluster the data into small groups using Kmeans with a distance defined as their temporal correlations. The results show K-means clustering matched 97.2% with the transmission zone assignments. The second step is to train an LSTM-Autoencoder model for each cluster leveraging both temporal and spatial correlations of the LMP inputs. The trained model is then used to identify anomalies caused by mixing data from other clusters to the target cluster. The identified anomaly nodes indicate the potential attack location. Since it is infeasible to obtain PJM cyber-attack data, we used data that deviated from the "cluster behavior" to mimic the attack impact when new congestion patterns are introduced. We provide also the procedure to search for the best hyperparameters to achieve the best performance. Further testing of the locational detection are carried out and evaluated on the simulation datasets with actual cyber-attacks.

4.5 Real-time Locational Anomaly Detection - Part II Simulation Dataset

In this section, we provide the evaluation of the locational anomaly detection algorithms on the simulation dataset. The objective is to identify anomalous pricing deviations which can possibly be attacks and localize the regions of affected nodes.

4.5.1 Data Overview

As described in Chapter 3, each generated dataset contained 170 days of data at five minute increments and in total 48960 data points. Figure 4.37 shows and example of 1 day (288 samples – 5 min intervals) of LMP data for all 39 buses with no attacks. Figure 4.38 shows an example for a duration of 1 day of LMP data with 10 cyber-attacks events. Dashed lines indicate attack times.

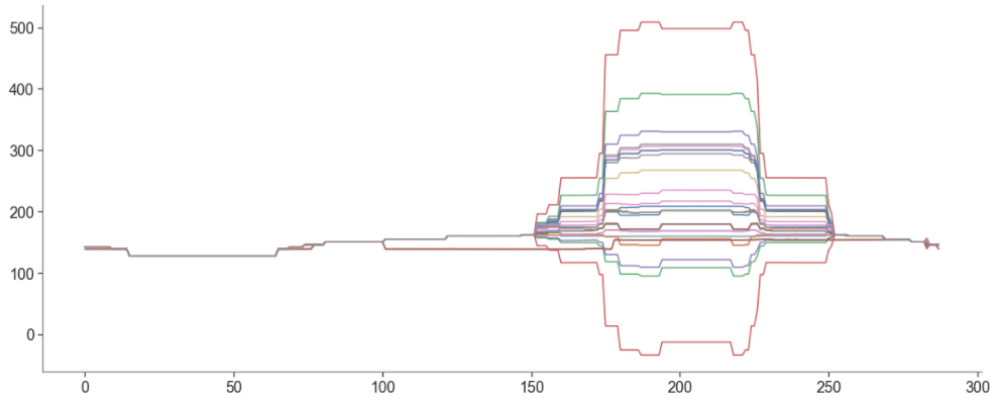


Figure 4.37: 39 bus system – 1 day of LMP ($/MW) data with no attacks.

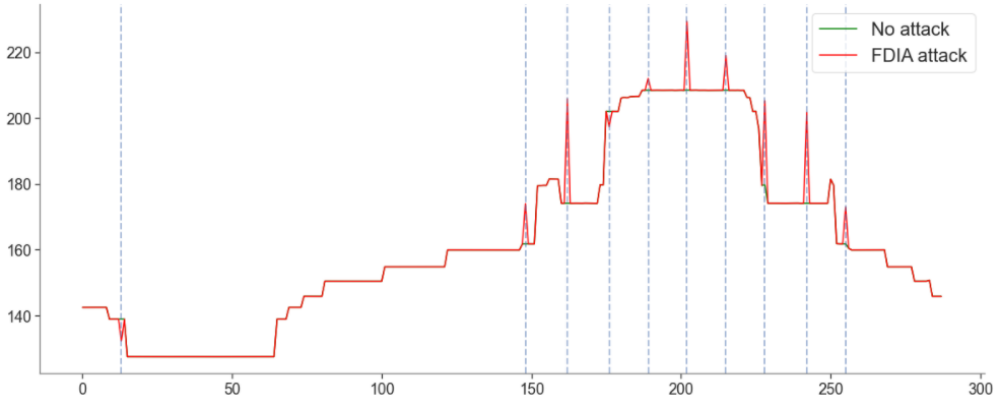


Figure 4.38: LMP ($/MW) data under 10 attack events.

4.5.2 K-means Clustering

Following the framework defined in Section 4.4, we first performed the correlation distance based K-means clustering. Correlations of LMP and its energy component, congestion component, and marginal loss component were explored over durations of day, week, 2 weeks, month, and 3 months. When exploring short durations of correlation they are strongly correlated and as the duration increases the correlations start to weaken. LMP's correlation tends to remain stronger for longer periods of time. Correlations of 39 nodes' LMPs over a month are plotted in Figure 4.39.

To confirm that the correlation between nodes does not change over time often we plotted the correlation between a reference node and 9 other nodes over a day (Figure 4.40). As shown, the correlation fluctuates over time but these fluctuations are similar with the other nodes, confirming the validity of this approach.

We then conducted the Elbow searching for the optimal number of clusters, shown in

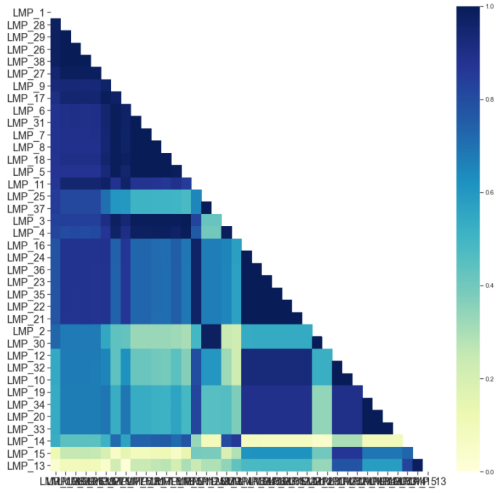


Figure 4.39: Correlation of all 39 buses over 1 month.

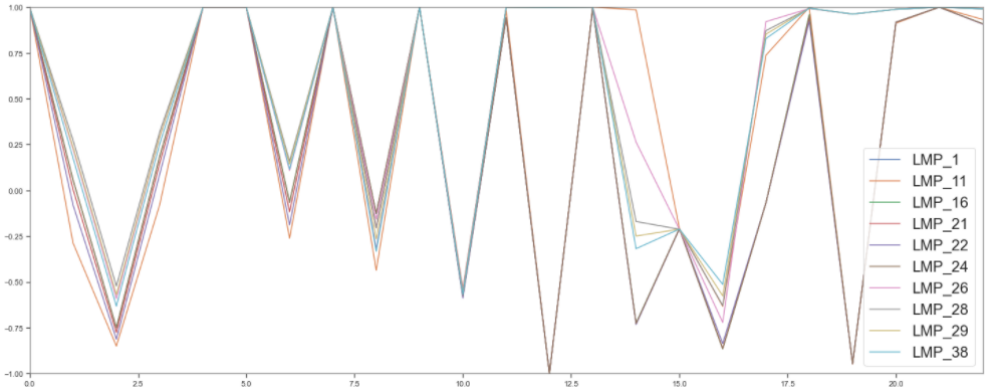


Figure 4.40: 10 correlations over one day.

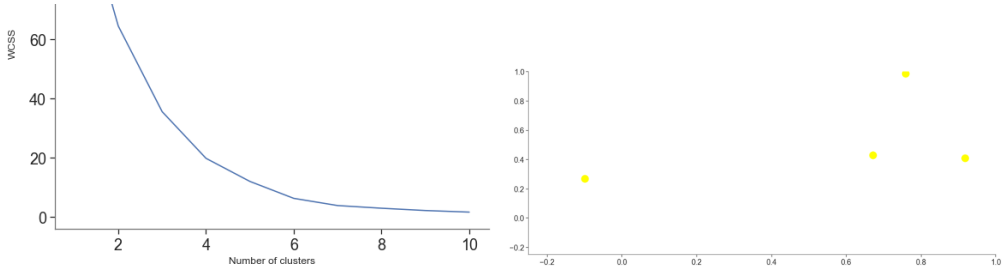


Figure 4.41: Elbow method optimization (left) and cluster centroids (right).

Figure 4.41.

Clustering results show the nodes in one cluster have strong correlations which ensures

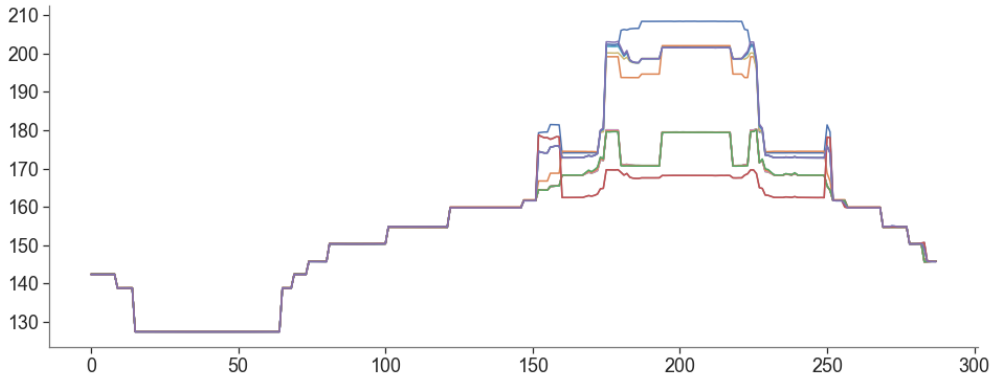


Figure 4.42: One day LMP data from a cluster with 16 nodes.

that the detection models will be able to capture the underlying group behavior. An example of one cluster with 16 nodes for a duration of one day is shown in Figure 4.42 and its correlations in Figure 4.43.

4.5.3 Data Scaling

Scaling is an essential data pre-processing step which is typically done by removing the mean and scaling to unit variance. However, outliers can often influence the sample mean / variance in a negative way. In such cases, the median and the interquartile range often give better results. This allows the models to converge much faster during gradient decent in the neural network learning process. We explored a few scalars for this task.

We tested our model using scalars from scikit learn's open source library: MinMaxScaler, RobustScaler, StandardScaler, MaxAbsScaler, QuantileTransformer, and PowerTransformer. Robust scalar produced the lowest RMSE (2.56), which was expected as it scales features using statistics that are robust to outliers. This Scalar removes the median and scales the data according to the quantile range defined as Interquartile Range (IQR). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile). Centering and scaling happen independently on each feature/node by computing the relevant statistics on the samples in the training set. Median and interquartile range are then stored to be used on later data using the transform method after predictions are done.

4.5.4 Parameter and Hyperparameter Optimization

The performance of neural network models largely depends on the parameters and hyperparameters which shape the network structure and determine its accuracy and validity. Model parameters are internal to the neural network for example, neuron weights. They are estimated or learned automatically from training samples. These parameters are also

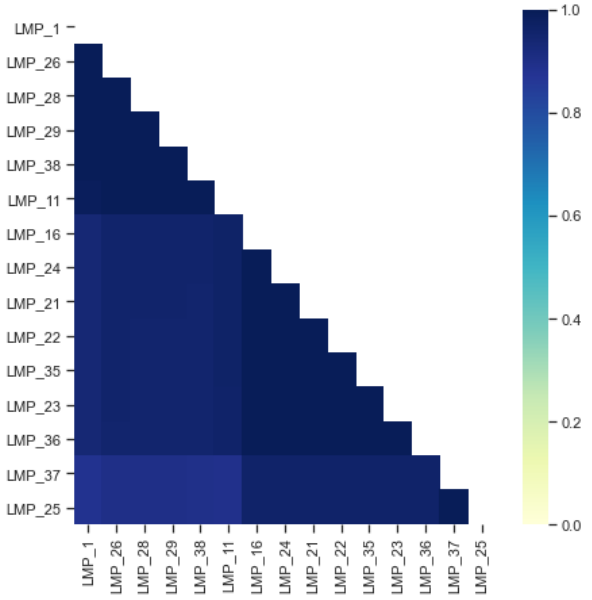


Figure 4.43: Correlation of a 16 node cluster.

used to make predictions. Hyperparameters are external parameters set by the operator of the neural network for example, selecting which activation function to use or the batch size used in training. There are various ways to optimize hyperparameters, from manual trial and error to sophisticated algorithmic methods, and there is no consensus on which works best. In this subsection, we introduce the process for hyperparameter tuning of the LSTM-Autoencoder model for simulation datasets.

Bottleneck Layer

In general, the bottleneck layer constrains the amount of information that goes through our auto-encoder, which forces the bottleneck to learn a "good but compressed" representation of our original input data. There is some work on how bottleneck size affects the overall quality of the embedding. In order to find the optimal number of neurons in the bottleneck layer, we tested 10 nodes as input and varied the bottleneck layer from 1-10 and measured the model's performance after training. We also examined the minimum training loss for each model. The optimal number of neurons for the bottleneck layer was determined to be 4. Figure 4.44 shows the minimum loss during training for one day data. On the x-axis the first number indicates the bottleneck number (1-4) and the second number indicates the batch size (64-128). This allows us to optimize both bottleneck layer and batch size simultaneously. This analysis was repeated with 1 week, 2 weeks, 1 month, 2 months and 3 months of data to ensure that the parameters can be used for different durations.

Activation Function

Activation function is the function through which we pass our weighted sum of the input, in order to have a significant output. The activation functions we tested were Sigmoid, Tanh

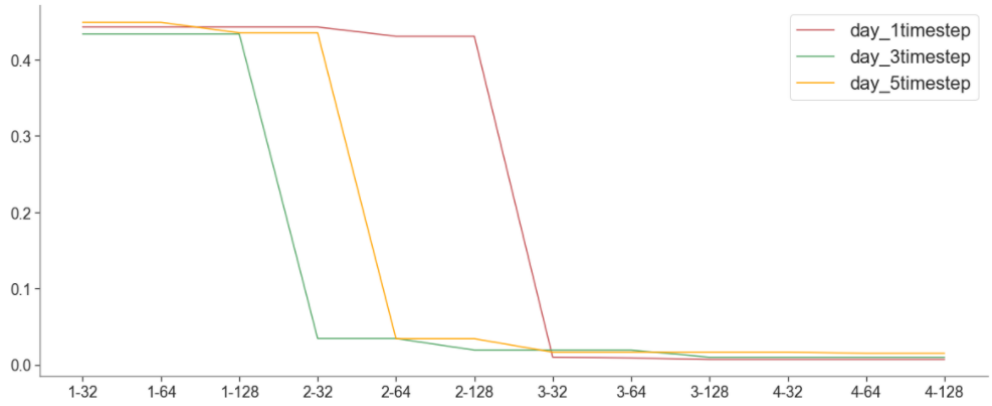


Figure 4.44: Losses minimum vs bottleneck layer.

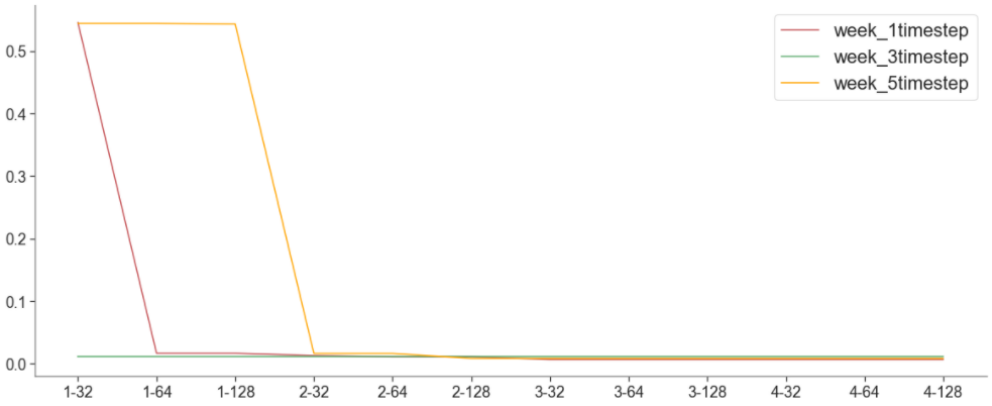


Figure 4.45: Losses minimum vs batch size.

and rectified linear unit (ReLU). The ReLU function is very quick in terms of training, and produced the best performing models in comparison to others.

Batch size

Batch size determines the frequency of updates. The smaller the batches, the more, and the quicker, the updates. The larger the batch size, the more accurate the gradient of the cost will be with respect to the parameters. That is, the direction of the update is most likely going down the local slope of the cost landscape. Having larger batch sizes, but not so large that they no longer fit in GPU memory, tends to improve parallelization efficiency and can accelerate training. Some publications argued that large batch sizes can hurt the model's ability to generalize by possibly causing the algorithm to find poorer local optima/plateau. Based on our results a batch size of 128 was chosen as the optimal size (Figure 4.45).

Epochs

Epochs represent the times that an algorithm trained on the whole dataset. The number of epochs depends on how the loss or error behaves for the training and validation data. As

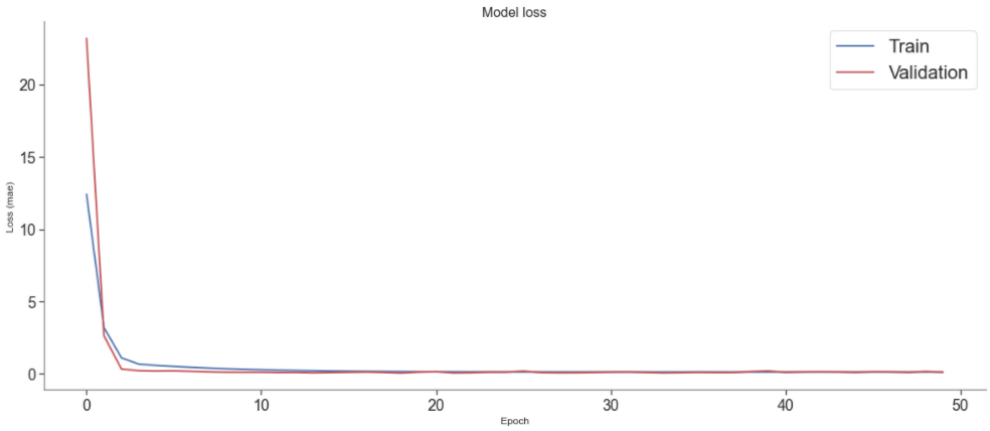


Figure 4.46: Losses MAE vs epochs during training.

long as it keeps dropping, training should continue. If the validation error starts increasing that might be an indication of over fitting. After experimenting with the different datasets (1 day, 1 week, etc.) it was shown 50 epochs (Figure 4.46) would be sufficient for training. We also implemented early stopping during training to avoid any over fitting.

Optimization Algorithm

The choice of optimizer influences both the speed of convergence and whether it occurs. Several alternatives to the classic gradient descent algorithms have been developed in the past few years. The Adam optimizer was chosen due to: 1) The hyperparameters of Adam (learning rate, exponential decay rates for the moment estimates, etc.) are usually set to predefined values, and do not need to be tuned. 2) Adam performs a form of learning rate annealing with adaptive step-sizes. 3) Adam compromises of both RMSProp and momentum. The downside of Adam is it uses the most memory for a given batch size in comparison to other optimizers. The loss function used was mean squared error.

Time Steps

LSTM takes into account the past data in addition to the current data in order to make "contextual" and more accurate predictions. Time steps is a parameter defining how many samples in the past we use to extract short term historical trend. We tested 1, 3 and 5 time steps as shown in Figure 4.47. In conclusion, we chose 1 time step as increasing the number of time steps did not improve the model's performance.

4.5.5 LSTM-Autoencoder for anomalous cluster detection

After fine tuning the hyperparameters, we trained an LSTM-Autoencoder model for each cluster. The LSTM-Autoencoder model outputs a reconstruction error for each signal under test. We first experimented the capability of LSTM-Autoencoder model in identifying anomalous conditions where the averaged reconstruction error of the cluster goes beyond the

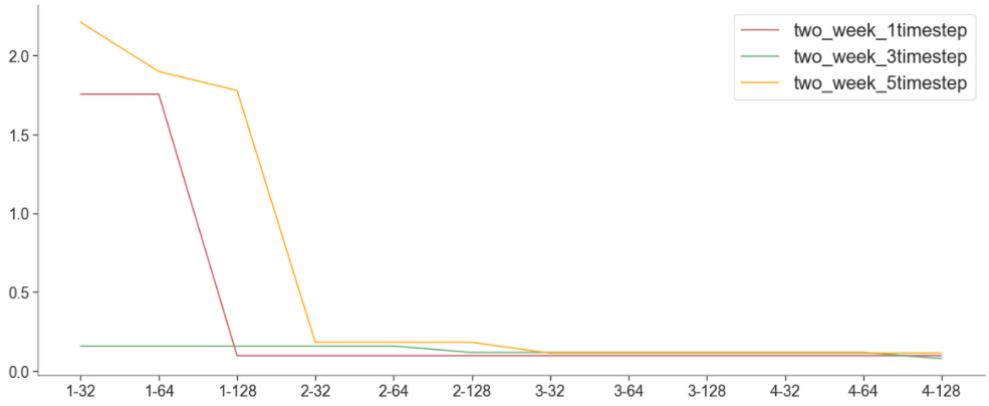


Figure 4.47: Losses minimum vs time steps.

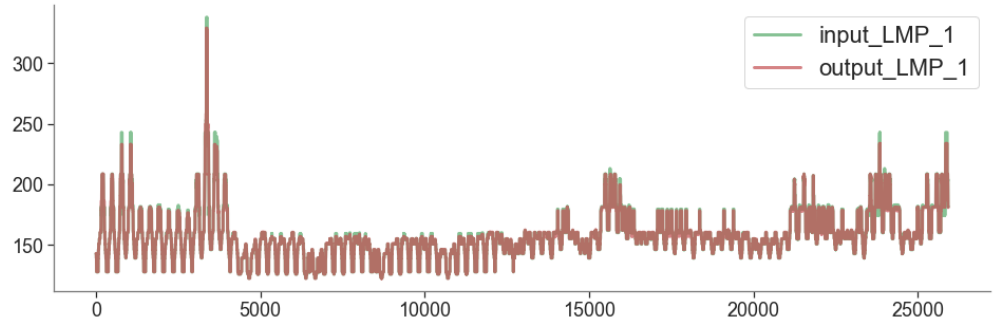


Figure 4.48: Training LMP input/output example node – 3 months.

threshold.

Threshold Optimization

To search for the optimal detection threshold, we need to first create a separate test dataset for iterative testing under different thresholds. The data set was split into 4 subsets: training data, training threshold data, testing threshold data and testing data. For each subset we evaluated the reconstruction errors and for the testing threshold set we tested at different threshold values while aiming for the highest detection rate and lowest false alarm rate. Other evaluation metrics were computed as well such as accuracy, recall, precision and F1 score.

Training data set:

Using the training set, we evaluated the model's ability to fit the none-attacked data and examined the reconstruction errors in order to form a baseline value for a threshold. An example of a node's LMP from the training set is in Figure 4.48.

Reconstruction errors peaked at 12 and 95 percent of them fell below 2.5 as shown in Figure 4.49.

Training threshold dataset:

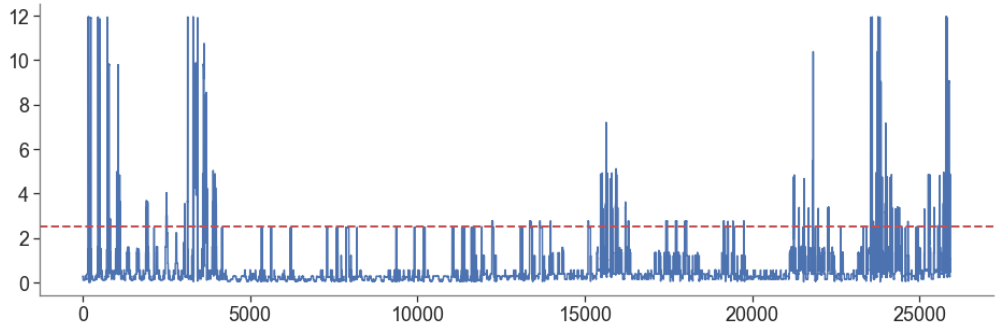


Figure 4.49: Reconstruction error of the training data in Figure 4.48.

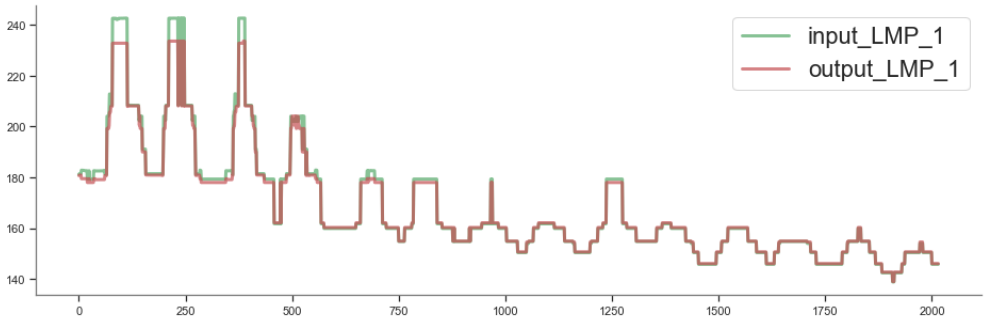


Figure 4.50: Training threshold LMP input/output example node – 1 week.

From the training threshold set, we performed the same steps we did with the training data. We evaluated the model's ability to fit the none-attacked data and examined the reconstruction errors in order to form a second baseline value for a threshold. An example of a node's LMP from the training set is in Figure 4.50.

Reconstruction errors peaked at 10 and 95 percentage of them fell below 4.9 as shown in Figure 4.51. From both the training set and training threshold sets we have an approximation threshold value between 2.5 and 4.9.

Testing threshold dataset:
Using the testing threshold set, we evaluated the model's ability to fit the attacked data and evaluate the reconstruction errors. An example of a node's LMP from the testing threshold set is shown in Figure 4.52. The blue vertical lines indicate the time when attacks occurred.

Reconstruction errors peaked at 15 and 95 percent fell below 5 as shown in Figure 4.53. From both the training set, training threshold and testing threshold sets we have an approximate threshold value of 5. We tested our model with values below and above the 5 baseline and evaluated the DR and FAR metrics. Based on the results shown in Table 4.16 we observe that a threshold of 1 produces the best results when the target is to keep a high detection rate.

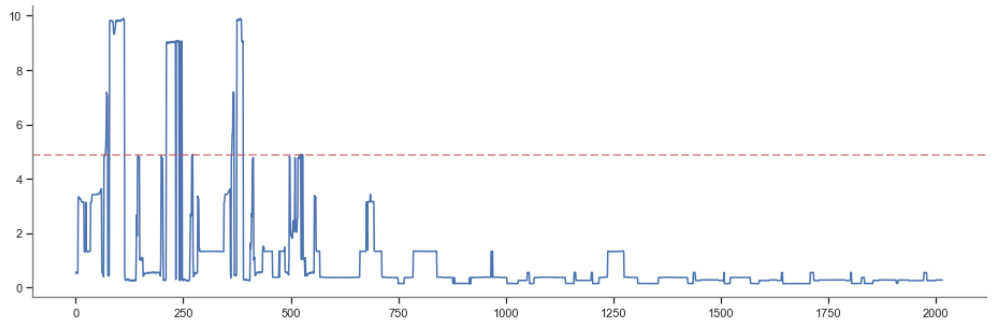


Figure 4.51: Reconstruction error of the training threshold data in Figure 4.50.

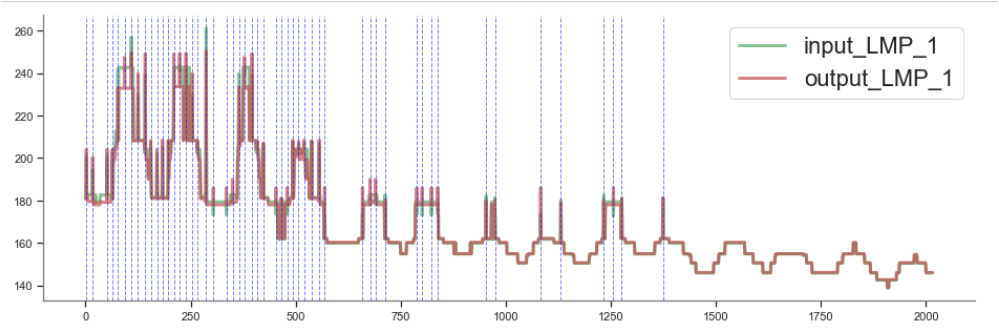


Figure 4.52: Testing threshold LMP input/output example node – 1 week.

We next performed a refined searching near 1 using decimal thresholds. The decimal thresholds and their model performance are listed in Table 4.17. Based on the larger difference between DR and FAR, a 0.9 threshold value is considered the optimal value.

Testing data set:

Using the selected threshold, we then evaluated the model's ability to fit the attacked data and examined the reconstruction errors. An example of a node's LMP from the testing set is in Figure 4.54. The blue vertical lines indicate the time when attacks occurred.

Reconstruction errors peaked at 14 and the optimal threshold was set to 0.9 as shown in Figure 4.55. The model's DR and FAR metrics under 0.9 threshold are considered to be the final performance. Based on the results shown in Table 4.18 we see that a threshold of 0.9 does produce the best results in comparison to the other threshold values.

Model Performance

The final anomaly detection performance for the LSTM-Autoencoder model has a detection rate of 100% and a false alarm rate of 1.7%. The testing procedure runs on a laptop (Intel Core i7-6820HQ CPU 2.7GHz) for a total of 40320 data points in 250 ms that is 6.2 microseconds per detection.

The high DR indicates the reconstruction errors can accurately reflect the impact from cyber-attacks. For real-world implementation, we will use LSTM-Autoencoder models trained

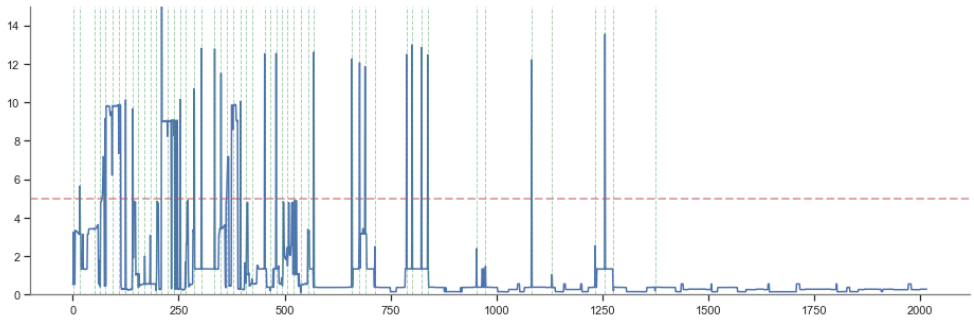


Figure 4.53: Reconstruction error of the testing threshold data in Figure 4.52.

Threshold	Accuracy	TP	TN	FN	FP	Recall	Percision	F1	DR	FAR
1	0.833135	441	16355	79	3285	0.848077	0.118357	0.207725	0.848077	0.167261
2	0.890129	357	17588	163	2052	0.686538	0.148194	0.243769	0.686538	0.104481
3	0.923313	277	18337	243	1303	0.532692	0.175316	0.263810	0.532692	0.066344
4	0.937897	240	18668	280	972	0.461538	0.19802	0.277136	0.461538	0.049491
5	0.942659	219	18785	301	855	0.421154	0.203911	0.274780	0.421154	0.043534
6	0.951984	203	18989	317	651	0.390385	0.237705	0.295488	0.390385	0.033147
7	0.955556	177	19087	343	553	0.340385	0.242466	0.283200	0.340385	0.028157
8	0.958383	157	19164	363	476	0.301923	0.248025	0.272333	0.301923	0.024236
9	0.96002	149	19205	371	435	0.286538	0.255137	0.269928	0.286538	0.022149
10	0.960913	137	19235	383	405	0.263462	0.252768	0.258004	0.263462	0.020621
11	0.965427	119	19344	401	296	0.228846	0.286747	0.254545	0.228846	0.015071

Table 4.16: Integer value threshold model evaluation.

Threshold	Accuracy	TP	TN	FN	FP	Recall	Percision	F1	DR	FAR
0.1	0.272173	510	4977	10	14663	0.980769	0.033612	0.064997	0.980769	0.746589
0.2	0.487004	496	9322	24	10318	0.953846	0.045866	0.087524	0.953846	0.525356
0.3	0.532887	495	10248	25	9392	0.951923	0.050066	0.095128	0.951923	0.478208
0.4	0.627629	490	12163	30	7477	0.942308	0.061504	0.115471	0.942308	0.380703
0.5	0.674752	478	13125	42	6515	0.919231	0.068354	0.127246	0.919231	0.331721
0.6	0.713938	470	13923	50	5717	0.903846	0.075966	0.140152	0.903846	0.29109
0.7	0.751885	466	14692	54	4948	0.896154	0.086073	0.157061	0.896154	0.251935
0.8	0.759425	459	14851	61	4789	0.882692	0.087462	0.159154	0.882692	0.243839
0.9	0.819395	451	16068	69	3572	0.867308	0.112105	0.198547	0.867308	0.181874
1	0.833135	441	16355	79	3285	0.848077	0.118357	0.207725	0.848077	0.167261

Table 4.17: Decimal value threshold model evaluation for threshold test data.

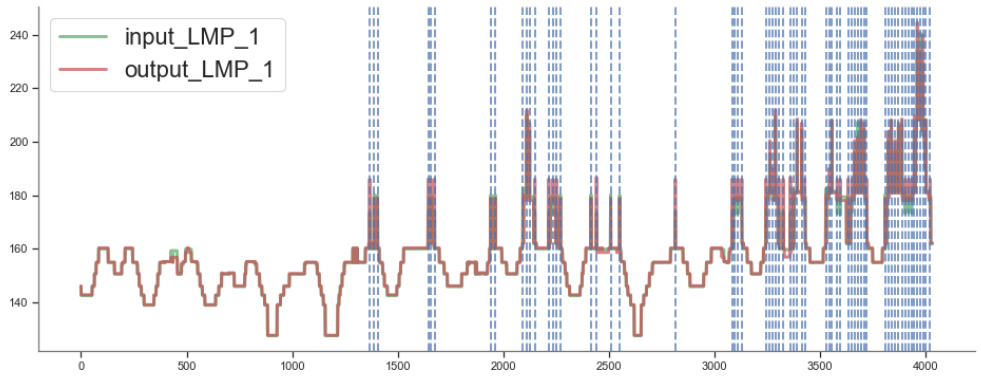


Figure 4.54: Testing threshold LMP input/output example node – 2 weeks.

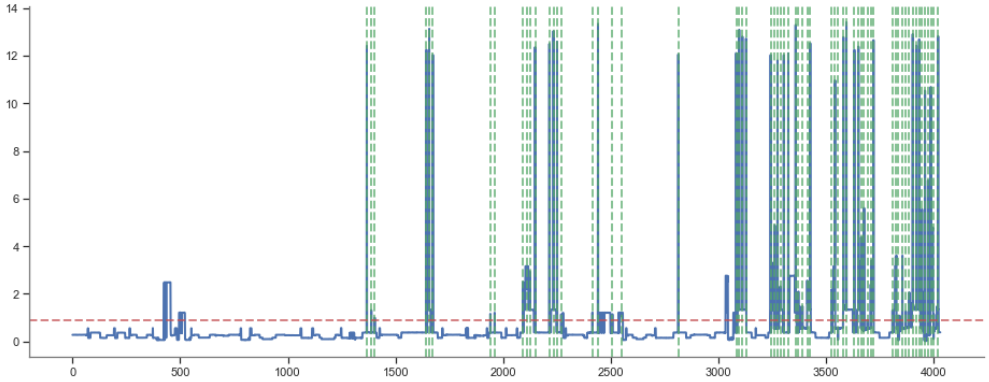


Figure 4.55: Reconstruction error of the testing data in Figure 4.54.

Threshold	Accuracy	TP	TN	FN	FP	Recall	Percision	F1	DR	FAR
0.1	0.395089	63	1530	0	2439	1	0.02518	0.049123	1	0.614512
0.2	0.647073	63	2546	0	1423	1	0.042396	0.081343	1	0.358529
0.3	0.740575	63	2923	0	1046	1	0.056808	0.107509	1	0.263542
0.4	0.944692	63	3746	0	223	1	0.22028	0.361032	1	0.056185
0.5	0.947421	63	3757	0	212	1	0.229091	0.372781	1	0.053414
0.6	0.947421	63	3757	0	212	1	0.229091	0.372781	1	0.053414
0.7	0.978671	63	3883	0	86	1	0.422819	0.59434	1	0.021668
0.8	0.982391	63	3898	0	71	1	0.470149	0.639594	1	0.017889
0.9	0.982391	63	3898	0	71	1	0.470149	0.639594	1	0.017889
1	0.982143	62	3898	1	71	0.984127	0.466165	0.632653	0.984127	0.017889

Table 4.18: Decimal value threshold model evaluation for threshold test data.

for each cluster to detect anomalous clusters and these clusters point to the attack region. Meanwhile, the LSTM-Autoencoder model generates a reconstruction error for LMP at each node. By ranking these errors, we get a list of highly impacted nodes which is used to localize the attack region within the cluster. This is discussed in the next subsection.

4.5.6 LSTM-Autoencoder for in-cluster localization

In this subsection, we further tested the model's capability of localizing the most affected nodes inside one cluster. The idea is to refine the search of the attack region after one cluster is identified as an anomalous group. For localization tests, we generated datasets with FDIA at different transmission lines in the IEEE 39Bus system and datasets with LRA at different buses in the system. Since the 39Bus system is a small system, we used all 39 buses as one cluster and trained one LSTM-Autoencoder model for this cluster. In the following tests:

1) Reconstruction errors were analyzed only when the model detects an attack.

2) All reconstruction errors were normalized by the number of attacks per bus.

3) Reconstruction errors were ranked from highest value to lowest to determine the most affected buses in the system.

Localizing FDIA

In order to evaluate the model's localization capability with FDIA, 46 datasets were generated with each one attacking a different line of the system. Results of top 5 most affected buses are shown in Table 4.19 with Column "1" as the most impacted bus. Each row represents the FDIA attacks applied on one certain transmission line noted as "L# Attack" where # denotes the line number. Note that any lines that did not have more than 4 attacks were not included in Table 4.19(i.e., L2 attack)

Location	**1**	**2**	**3**	**4**	**5**
L1 Attack	LMP_9	LMP_7	LMP_6	LMP_1	LMP_27
	0.7628	0.7264	0.5397	0.4958	0.4395
L3 Attack	LMP_7	LMP_27	LMP_24	LMP_17	LMP_13
	0.7337	0.5153	0.5049	0.4846	0.4816
L4 Attack	LMP_7	LMP_27	LMP_17	LMP_26	LMP_6
	0.8765	0.8506	0.7195	0.7152	0.6756
L5 Attack	LMP_7	LMP_27	LMP_17	LMP_13	LMP_24
	0.7629	0.508	0.4919	0.4716	0.4539
L6 Attack	LMP_7	LMP_27	LMP_6	LMP_31	LMP_19
	0.7579	0.5564	0.5149	0.463	0.4565
L7 Attack	LMP_7	LMP_8	LMP_19	LMP_34	LMP_20
	0.5998	0.4	0.3814	0.3644	0.3626

Location	1	2	3	4	5
L8 Attack	LMP_7	LMP_9	LMP_8	LMP_19	LMP_34
	0.7377	0.564	0.5297	0.4738	0.4686
L9 Attack	LMP_7	LMP_27	LMP_6	LMP_17	LMP_24
	0.7974	0.7026	0.6398	0.6117	0.5646
L10 Attack	LMP_7	LMP_19	LMP_34	LMP_20	LMP_33
	0.6441	0.5705	0.5655	0.5606	0.5536
L11 Attack	LMP_4	LMP_14	LMP_3	LMP_13	LMP_32
	1.9954	1.904	1.8526	1.7847	1.7302
L12 Attack	LMP_9	LMP_1	LMP_4	LMP_8	LMP_14
	2.9205	2.4487	2.1578	2.132	2.1301
L13 Attack	LMP_7	LMP_6	LMP_31	LMP_8	LMP_5
	12.2221	11.5013	10.9561	9.2717	5.913
L15 Attack	LMP_3	LMP_7	LMP_5	LMP_4	LMP_8
	1.3115	1.2659	1.0414	0.987	0.9344
L16 Attack	LMP_4	LMP_3	LMP_5	LMP_8	LMP_14
	2.529	2.3346	2.2972	2.2802	2.1896
L18 Attack	LMP_6	LMP_31	LMP_7	LMP_11	LMP_8
	12.5608	12.3309	11.6722	9.8777	8.0906
L19 Attack	LMP_7	LMP_6	LMP_31	LMP_8	LMP_18
	21.6225	19.8463	16.2226	13.9078	13.7014
L20 Attack	LMP_7	LMP_8	LMP_9	LMP_5	LMP_6
	3.2338	2.6391	2.5269	2.4952	2.4574
L23 Attack	LMP_27	LMP_4	LMP_3	LMP_14	LMP_13
	1.6242	1.5567	1.4399	1.3356	1.288
L24 Attack	LMP_3	LMP_27	LMP_5	LMP_4	LMP_26
	3.1348	2.7685	2.6388	2.5471	2.5372
L25 Attack	LMP_1	LMP_9	LMP_30	LMP_2	LMP_25
	5.3041	5.0115	3.9135	3.8978	3.343
L27 Attack	LMP_9	LMP_1	LMP_3	LMP_26	LMP_2
	1.7024	1.4789	1.3091	1.0965	1.0549
L28 Attack	LMP_7	LMP_8	LMP_6	LMP_31	LMP_9
	2.168	2.0847	1.9083	1.8142	1.8013
L29 Attack	LMP_7	LMP_27	LMP_8	LMP_5	LMP_3
	1.7363	1.616	1.3814	1.3431	1.3362
L30 Attack	LMP_9	LMP_8	LMP_4	LMP_14	LMP_5
	2.9417	2.8817	2.7519	2.6314	2.6113
L31 Attack	LMP_27	LMP_7	LMP_6	LMP_8	LMP_31
	2.5213	1.7568	1.6869	1.5905	1.5074
L32 Attack	LMP_7	LMP_27	LMP_8	LMP_9	LMP_31

Location	1	2	3	4	5
	2.0886	2.004	1.8923	1.8012	1.7405
L33 Attack	LMP_7	LMP_8	LMP_3	LMP_9	LMP_5
	2.7195	2.3938	2.3813	2.3706	2.3141
L35 Attack	LMP_7	LMP_6	LMP_31	LMP_27	LMP_18
	12.6318	11.1151	10.3826	10.0879	8.6492
L36 Attack	LMP_6	LMP_31	LMP_9	LMP_1	LMP_8
	2.737	2.3616	2.3021	2.165	2.0631
L37 Attack	LMP_9	LMP_1	LMP_2	LMP_6	LMP_30
	2.4946	2.0678	1.6222	1.5705	1.5325
L38 Attack	LMP_9	LMP_7	LMP_1	LMP_3	LMP_8
	3.1097	2.3357	2.2658	2.0941	1.9516
L39 Attack	LMP_8	LMP_7	LMP_5	LMP_9	LMP_4
	2.6155	2.6049	2.4855	2.4594	2.4265
L40 Attack	LMP_26	LMP_28	LMP_29	LMP_38	LMP_27
	8.6589	7.3645	6.8644	5.9959	5.6945
L41 Attack	LMP_8	LMP_3	LMP_5	LMP_7	LMP_4
	1.6447	1.581	1.5514	1.4883	1.4679
L42 Attack	LMP_6	LMP_9	LMP_1	LMP_31	LMP_8
	2.9191	2.6578	2.5651	2.5482	2.2178
L43 Attack	LMP_9	LMP_1	LMP_3	LMP_2	LMP_30
	3.2057	3.0527	2.1639	2.1607	2.1507
L45 Attack	LMP_26	LMP_28	LMP_29	LMP_27	LMP_38
	105.36	96.9464	89.568	83.9417	79.0562
L46 Attack	LMP_26	LMP_28	LMP_29	LMP_38	LMP_27
	97.6362	85.4361	79.9477	70.4871	66.2189

Table 4.19: Results of top 5 most affected buses for FDIA attacks with average reconstruction errors.

Form the first 7 rows on Table 4.19, we identified three major attack regions, marked on the topology diagram in Figure 4.56. The red region is a heavy load area linked to FDIA applied on lines that transfer power into the area. The green region is another load area and the yellow region is a local generation zone. The attacks applied on the transmission lines that deliver power from generator 8 are largely affecting the green region. L10 attack which blocks the power flowing from the yellow region to the red region affects both areas. Overall, the red region is the most sensitive region to most of the attacks since the FDIA creates intentional congestion in the system and the load supply might be interrupted. Two observations can be drawn from the data: (1) the most sensitive nodes might not be the direct attack target; (2) the distribution of the sensitive nodes are guided by the power

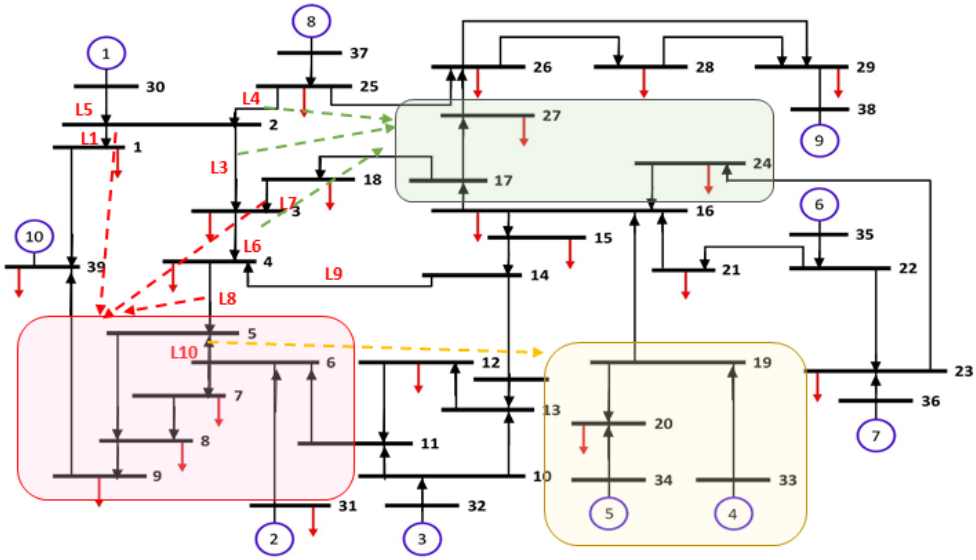


Figure 4.56: FDIA attacks mapped to the attack regions on IEEE 39Bus system.

delivery flow. By searching in the direction of the power flow, operators can find the nearby region of the attack target.

Localizing LRA

To evaluate the localization performance with LRA, 39 datasets were generated with each one having attacks at a different bus in the system. Results of top 5 most affected buses are shown in Table 4.20. The format of the table and its description is similar to Table 4.19. Note that any buses that did not have more than 4 attacks were not included in the table (i.e., BUS2 attack).

The LRA test data point to the same three attack regions discussed above. Similar with FDIA, LRA also introduced local congestion in the system. It is more obvious from Table 4.20 that the LRA attacks are mapped to their closest region.

4.5.7 Conclusion

In this section, we presented the procedure of using LSTM-Autoencoder model to identify region of interest. Specifically, we first studied the correlation behavior and performed K-means clustering for the simulation dataset. We then leveraged the cluster behavior to detect anomalies by comparing the reconstruction error with an optimized threshold. The detector shows 100% detection for clusters that contain cyber attacks, which achieves the Milestone goal of 85% detection of attack regions. Further, we explained how the model can localize the inner cluster sensitive nodes. These nodes can be mapped to potential attack regions when traced with the power flow directions.

Location	1	2	3	4	5
BUS 1 Attack	LMP_9	LMP_4	LMP_34	LMP_19	LMP_5
	2.2948	2.0012	1.8339	1.8244	1.5951
BUS 3 Attack	LMP_9	LMP_5	LMP_4	LMP_7	LMP_27
	3.6723	3.5212	3.4931	3.0361	2.6855
BUS 4 Attack	LMP_9	LMP_5	LMP_4	LMP_7	LMP_8
	9.1769	8.9522	8.9122	6.8857	6.6215
BUS 7 Attack	LMP_7	LMP_5	LMP_27	LMP_9	LMP_11
	2.6983	2.6772	2.5558	2.0859	1.9764
BUS 8 Attack	LMP_7	LMP_5	LMP_27	LMP_9	LMP_11
	2.3226	2.2751	2.1623	1.9334	1.6931
BUS 9 Attack	LMP_5	LMP_7	LMP_11	LMP_27	LMP_6
	2.4363	2.4136	2.3362	2.2716	2.1779
BUS 12 Attack	LMP_34	LMP_19	LMP_9	LMP_33	LMP_27
	7.1756	7.1646	6.7976	6.5123	6.5024
BUS 15 Attack	LMP_19	LMP_34	LMP_27	LMP_33	LMP_9
	8.4318	8.4202	7.9931	7.6859	7.4392
BUS 16 Attack	LMP_27	LMP_34	LMP_19	LMP_9	LMP_33
	13.1747	11.2889	11.2474	10.6298	10.2714
BUS 18 Attacks	LMP_7	LMP_5	LMP_9	LMP_27	LMP_3
	2.238	2.1977	2.033	1.9421	1.6301
BUS 20 Attack	LMP_34	LMP_19	LMP_33	LMP_27	LMP_9
	10.6361	10.5494	9.579	9.4834	8.7676
BUS 21 Attack	LMP_19	LMP_34	LMP_33	LMP_4	LMP_20
	8.5072	8.3674	7.6176	7.5254	6.5798
BUS 23 Attack	LMP_4	LMP_19	LMP_34	LMP_33	LMP_20
	7.7493	7.5597	7.4229	6.7545	5.9138
BUS 24 Attack	LMP_27	LMP_34	LMP_19	LMP_9	LMP_33
	11.6467	11.2194	11.1975	10.6228	10.0664
BUS 25 Attack	LMP_9	LMP_4	LMP_26	LMP_34	LMP_19
	2.5674	2.0324	1.8103	1.7655	1.7643
BUS 26 Attack	LMP_9	LMP_4	LMP_19	LMP_34	LMP_26
	1.7228	1.7094	1.6965	1.6234	1.4872
BUS 27 Attack	LMP_19	LMP_34	LMP_4	LMP_9	LMP_33
	1.6988	1.6274	1.5648	1.544	1.3203
BUS 28 Attack	LMP_19	LMP_34	LMP_33	LMP_4	LMP_20
	2.4056	2.345	2.0534	1.9187	1.8242
BUS 29 Attacks	LMP_19	LMP_34	LMP_4	LMP_33	LMP_9
	2.2974	2.2631	2.102	1.9551	1.7936
BUS 31 Attack	LMP_7	LMP_5	LMP_27	LMP_9	LMP_11
	3.3061	3.1231	2.6084	2.5574	2.4745
BUS 39 Attack	LMP_9	LMP_5	LMP_7	LMP_3	LMP_4
	1.8263	1.8212	1.8112	1.3544	1.2875

Table 4.20: Results of top 5 most affected buses for LRA attacks with average reconstruction errors.

Table 4.21: Statistics of the PJM LMP dataset.

Stats	LMP	Energy Price	Congestion Price	Marginal Loss Price
Mean	23.70	27.14	-3.11	-3.19
Standard deviation	58.61	54.38	27.94	2.64
Minimum	-580.11	4.44	-3.97	-317.22
Maximum	3,442.86	2,361.2	1,656.72	4,367.60

4.6 Price Spike Anomaly Detection

Price spikes are the short and sharp fluctuations in electricity prices, generally caused by the sudden occurrence of imbalance between demand and supply. Due to its discontinuity and uncertainty, price spikes often cause high prediction/reconstruction errors in anomaly detection leading to false positives. To address this issue, in this section, we performed two studies: (1) pattern identification based price spike detection tested on PJM dataset and (2) classification-based price spike detection tested on ISO-NE data.

4.6.1 Pattern identification of LMP spikes

In this subsection, we analyzed the patterns of spikes in the energy component of locational marginal price (LMP) in PJM data.

Basic Statistics of Spikes

This PJM dataset contains five-minute interval of real-time market data from September 1, 2019 to October 1, 2019. It contains 23,121,223 records for 2810 pnode IDs and 1408 pnode names, where a pnode is used by PJM to denote a pricing node. The basic statistics of the dataset is shown in Table 4.21. Note that the highest LMP can exceed $3400 in the PJM real-time market, which is in sharp contrast to the average value of $23 per MWh.

In order to understand patterns of energy spikes, we examined the location where the highest energy spike is observed in the entire dataset. Figure 4.57 shows the LMP data of this particular pnode. Following [86], we employed a threshold of $2000 for spikes. Namely, a spike is recorded if the LMP is greater or equal to the threshold. Under this definition, there are 4399 spikes among the PJM dataset. In particular, the spikes are observed in a wide spread of locations, 2410 pnodes, or 86% of all pnodes. On the other hand, the spikes highly concentrate in time, 6 timestamps or 0.069% of all timestamps. These observations indicate if a spike pattern is not in the observed time windows, then it is highly likely to be due to an attack and if a spike pattern is sparse in a few locations, then it is also likely to be an attack.

Spike Types

We next considered the characteristics of spikes in order to categorize them in different types,

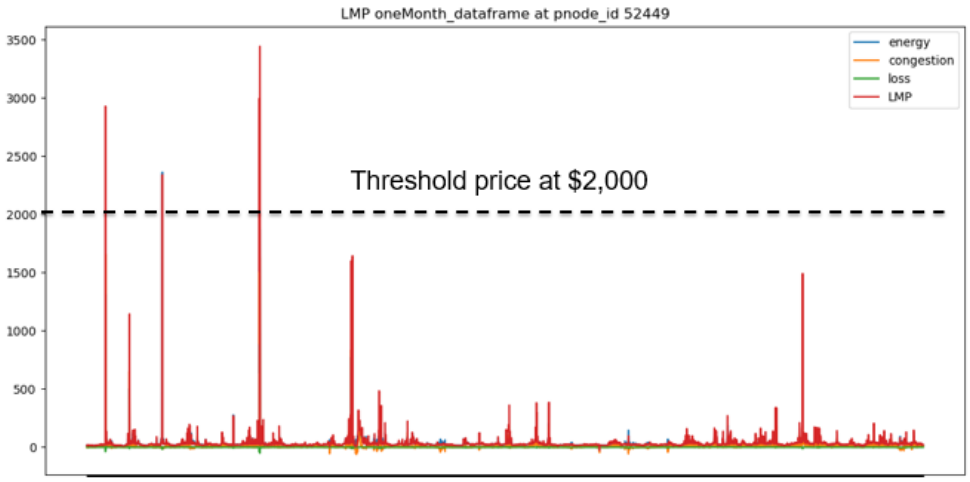


Figure 4.57: LMP data of the pnode with the highest LMP

namely, short spikes, sustained spikes, and camel spikes:

- Short: Only one spike index in the window
- Sustained: At least two consecutive spike indices in the window
- Camel: Two spike indices in the window with non-spike index in between

Figure 4.58 shows examples of the three types of spikes. If a spike lasts only one timestamp (i.e., five minutes), then it is a short spike. If a spike lasts two or more consecutive timestamps (i.e., 10 minutes or more), then it is a sustained spike. Finally, a camel spike is a spike that rises above the threshold, then drops below the threshold, and then rises above the threshold again. The size of the window for the spike characteristics was chosen to be 10 timestamps before and after the first value that exceeds the threshold.

Under this definition, among the 4399 spikes, there are 196 short spikes, 197 camel spikes, and 4006 sustained spikes. Table 4.22 categorizes the number of spikes with respect to the timestamps. Note that there is overlap between sustained and camel spikes at timestamp 2019-09-07 15:20:00 and 2019-09-07 15:25:00. So the number of sustained spikes is calculated as $1793 \check{} 197 + 2410 = 4006$.

4.6.2 Temporal and Spatial Characteristics

Time Instance and Location

As previously mentioned, the LMP spikes are concentrated in time and yet wide spread in location; see Table 4.23. In particular, a spike with LMP greater than $2000 are observed among 86% of 2810 locations in the PJM data. On the other hand, all these price spikes happen in a total of 6 time instances, which is only 0.069% over the one-month horizon

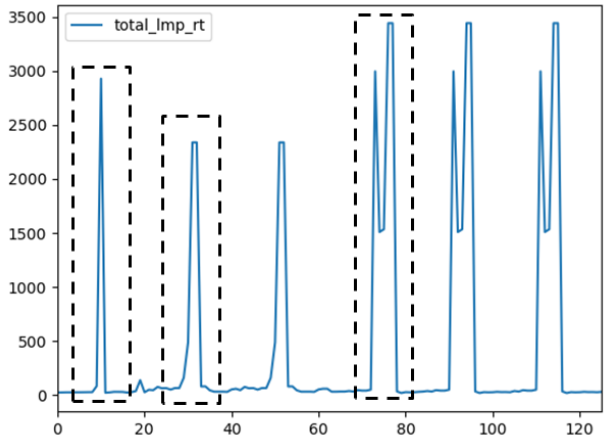


Figure 4.58: Three types of spike observed, short spikes, sustained spikes, and camel spikes

Table 4.22: Spike types and timestamps.

Timestamp	**Pnode_id**	**Type**
2019-09-03 09:40:00	196	short
2019-09-07 15:05:00	197	camel
2019-09-07 15:20:00	1793	camel
2019-09-07 15:25:00	1793	Sustained
2019-09-03 16:55:00	2410	sustained
2019-09-03 17:00:00	2410	sustained

data. To understand the spikes between different time instances, we examined the locations of the spikes and found that there is a progression of spike locations. For example, LMP spikes happen in a set of 196 pnodes on 2019-09-03 09:40:00, and in a set of 197 pnodes on 2019-09-07 15:05:00. The former set of pnodes is contained in the latter set of pnodes. This latter set of pnodes is then a subset of 1793 pnodes with LMP spikes on 2019-09-07 15:20:00 and 2019-09-07 15:25:00. This pattern continues, namely, the set of 1793 pnodes is a subset of 2410 pnodes with LMP spikes on 2019-09-03 16:55:00 and 2019-09-03 17:00:00. In other words, the PJM dataset displays a progression of spikes in location, that is, spikes observed in a smaller set of pnodes are to be observed in a bigger set of pnodes.

Table 4.23: Energy price spikes in time and location.

LMP >$2,000	**pnode_id**	**pnode_name**	**Time instance**
number of records	2410	926	6
percentage	(2410/2810) = 85.76%	(926/1408) = 65.76%	(6/288/30) = 0.069%

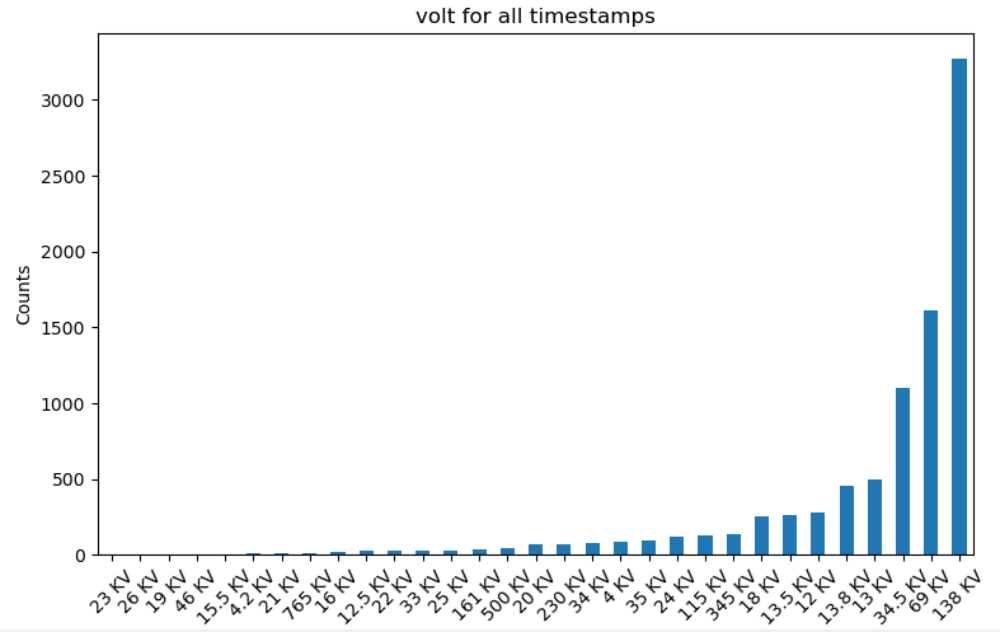


Figure 4.59: Distribution of LMP spikes among the buses with different voltages.

Spike Distribution

We next considered the voltage, types, zones, and geographic location of the spikes. The objective is to understand if there are patterns of spikes in other aspects of the system. Figure 4.59 shows the distribution of spikes among the buses with different voltages. Note that 138kV, 69kV, and 34.5kV buses rank as the buses with the three largest numbers of spikes, with a total of 67.5% spikes. In particular, we have 37% (138 KV) + 12.5% (34.5 KV) + 18% (69 KV) = 67.5%.

In terms of equipment, the majority of spikes, 82%, are observed at the load buses. 17% spikes are observed at the generation buses and less than 1% are observed at the extra high voltage buses.

For geographic distributions, the PJM dataset in this study consists of five zones, namely, Commonwealth Edison (ComEd), East Kentucky Power Cooperative (EKPC), Baltimore Gas and Electric Company (BGE), Dayton Power and Light Company (DPL), and PECO Energy. The distribution of spikes among these five zones is 58% for ComEd, 18% for EKPC, 14% for BGE, 5% for DPL, and 5% for PECO. Note that ComEd in the Chicago metropolitan area experiences a majority of the spikes. The metropolitan area with the next highest number of spikes is Baltimore (BGE) with 14% of spikes.

Detection of Abnormal Spikes

An anomaly detector was designed based on the temporal and spatial features of LMP spikes. In particular, if a price spike with a magnitude exceeding $2000 occurs outside of the timestamps in Table 4.22, or if it occurs outside of the 2410 pnodes, then it is highly likely

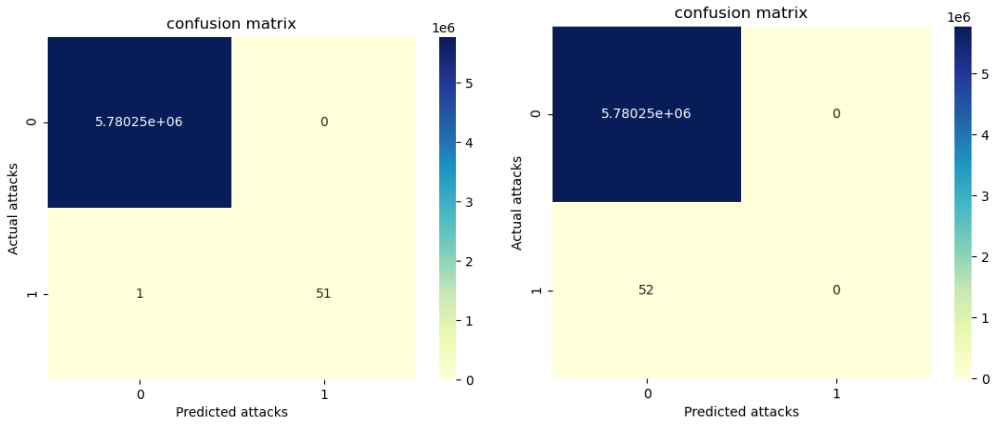


Figure 4.60: Confusion matrix of the logistic regression model for attacks over $2000 (left) and for attacks below $1500 (right).

to be an attack. Two scenarios were considered:

- Scenario 1: Randomly generate N attacks to LMP by changing LMP to values between $2000 and $3000, where N = 200, 2000, 20000.
- Scenario 2: Randomly generate N attacks to LMP by changing LMP to values between $1000 and $1500, where N = 200, 2000, 20000.

For each scenario, the spike, spike time, spike location were used as input features, and the attack label was the target. A standard logistic regression from Scikit-Learn was employed for the detection. By training the model with 75% of data and testing it on the remaining 25% of data, the model is able to differentiate 98% of attacks correctly from normal spikes over $2000. On the other hand, for attacks below $1500, the model fails to differentiate them. The confusion matrices for both scenarios are shown in Figure 4.60. In summary, the simple logistic detection based on the temporal and spatial features is capable of detecting random price attacks effectively.

4.6.3 Price Spike Analysis

In the following, we provide a survey of the literature on price spike prediction, a discussion on real world spike events and the price spike models based on PJM market rules.

Literature Review for Spike Prediction

Forecasting the electricity price is a well-studied topic in the literature. The prediction of spikes for energy price is quite different as it predicts the volatile spikes rather than the general price profiles. Recent years have seen approaches based on machine learning techniques to this challenging research problem. In [87], the authors developed a data mining

approach by using only market data. In particular, a Bayesian classifier was developed to predict spikes based on demand and supply of the marketplace. In [88], a hybrid model from wavelet and time domain data were proposed to forecast price spike occurrence. In [89], a stochastic regime switching model was proposed to predict spikes based on load and reserve margin. In [90], a number of machine learning approaches were compared for the price classification problem. It was shown that Bayes classifier outperforms decision tree for simulation results for the New York, Ontario, and Alberta electricity markets. In a recent report from Stanford ([86]), the authors showed that the gradient boosted classifier achieves 99.99% accuracy for predicting spikes in energy price for ISO New England data. In [91], an autoregressive conditional model was demonstrated to outperform memory-less models for spike forecasting. In [92], a support vector machine was developed for the forecast of price spikes in the electricity market. In [93], a comprehensive survey was provided for the electricity price forecasting.

Real World Events for Energy Price Spikes

Three real world events that caused energy price spikes and congestion in PJM area were reported in the media from 2015 to 2020. The first event happened on April 7, 2015, when the real-time energy prices in PJM market exceeded $500/MWh in BGE area and $400/MWh in Pepco area. The problem was traced back to an equipment failure in a transmission line in Pepco. As a result, 30,000 customers in Washington DC along with areas of Maryland were affected with a power outage ((Walton, PJM, New York electricity markets experience price spikes, 2015 [94]).

The second event happened on May 21, 2015, when energy prices in New York ISO's western zone jumped past $1300/MWh before 10am and fell back to $20/MWh within 30 minutes. The investigation suggested that the event was due to a change of congestion pattern with gas prices as the winter ended. In particular, the lower gas prices in Marcellus production area and more congestion in the eastern portion of New York led to a west-to-east power flow. While it was an isolated incident without any outages, the price spikes resulted in the rise of exports from NYISO to PJM [94].

The latest event happened on May 15, 2018, when the real-time energy price spiked above $600/MWh in PJM market around 4:50pm EDT. Specifically, the real-time price at Duke Energy Ohio Kentucky Zone spiked to $663.20/MWh and prices in most other PJM zones were $500/MWh. The cause of the spikes was that the system wide load reached 113,480 MW, which was more than 2% above the forecast peak at 110,650 MW. The temperature in Cincinnati set a record of 90 degrees Fahrenheit on May 14 (S&P Global Platts, 2018 [95]).

To understand the market mechanism that induces price spikes, two different market designs were considered: no price spike model and price spike model.

No Price Spike Model

PJM's price model is designed to rely on the capacity market to supply sufficient generation. This design aims to keep prices low (e.g., $150/MWh) during normal operation without price spikes. The market prices align with the system marginal cost, that is, the most expensive

power plant to be dispatched. The capacity market requires load serving entities to purchase approximately 20% more capacity than their peak load, otherwise subject to a financial penalty. While the threat of penalty induces load entities to purchase more generators, it also helps eliminate price spikes. This is possible only in a perfectly competitive market and the market is isolated (i.e., no import or export to other markets).

Rule 1: A capacity market that is isolated, perfectly competitive, and equipped with penalty mechanism does not induce price spikes in the long term operations. For example, from April 1, 1998 to April 1, 1999, no member company of PJM had market-based rates, so none of them could bid above their regulated marginal cost. When the PJM market exceeded the marginal cost, it was due to the market-based bidding of companies located outside of PJM [96].

Price Spike Model

The no-price-spike-model is an ideal model with simplified assumptions. In the real-world PJM data, the energy price spike could exceed 10 times the average cost. If the supply of generation is inadequate, the spike signals will induce investment in generation. The idea of a competitive market is to balance an optimal level of installed capacity and an optimal level of reliability. However, this purely market-based approach is almost always accompanied by regulations. The key requirement is the operating reserve margin. A higher level of the reserve margin leads to a higher level of installed capacity. Therefore, the installed capacity is determined by a market-driven process with regulatory inputs. The competitive market itself does not determine the required reserve margin.

Rule 2: An increase in reserve margin causes an increase in the level of installed capacity. While this rule makes intuitive sense, there is no direct link or mechanical connection between the reserve margin and the installed capacity. The process works through price spikes. In fact, the price spikes are largely determined by two non-market regulatory decisions: the operating reserve margin and the way the margin is enforced.

Rule 3: For the same installed capacity, price spikes occur sooner and last longer in a system with a higher reserve margin. For example, if the reserve margin is 5% and available reserves are 10%, then the system operator will not bid up prices of reserve. If the reserve margin is 12% and only 10% reserves are available, then the system operator will offer to pay a high price for energy.

Rule 4: The regulatory setting for pricing reserve purchases determines the shape of price spikes. If the rule is sharp, for example, pay up to $5000 if reserves fall below 5% and pay nothing if reserves are above 5%, then the price spike will jump sharply, but will have a short duration. On the other hand, if the rule setting increases gradually, then the price spike will be lower and broader.

Rule 5: Neither the height nor the width of the price spike induces investment in generation. The total area, that is, the profit, of the price spike induces investment.

The internal working of PJM on the regulatory setting of reserves is not public knowledge. Just as it is possible to set the wrong capacity requirement, so it is possible to set the wrong

Table 4.24: Statistics of LMP and its component.

	Energy	Congestion	Loss	LMP
MIN	-25.01	-227.92	-30.35	-150.00
MAX	354.05	632.35	18.28	673.26
MEAN	24.46	-0.02	-0.09	24.35

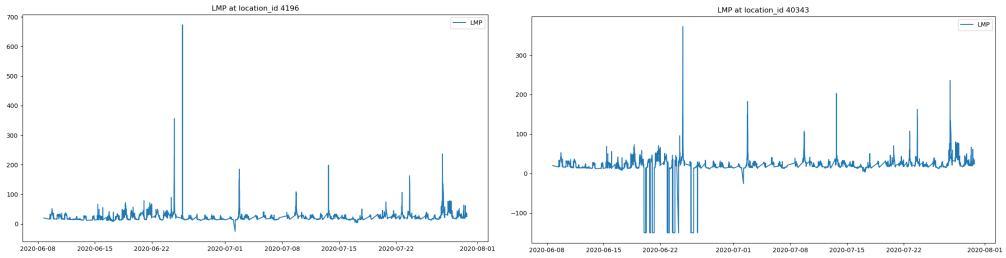


Figure 4.61: LMP with the highest (left) and the lowest (right) prices.

operating reserve requirement. Furthermore, the operating reserve approach has its effect by random variables such as weather conditions. Such randomness slows the process by which the market converges to its long-run equilibrium.

4.6.4 Classification-based Detection for ISO-NE data

Built upon our understanding on price spike patterns and mechanisms, we developed a series of classification based spike detectors which were tested on ISO-NE data for comparison with existing publications. The following subsection elaborates the feature selection and model testing results.

Five-Minute LMP Data

Our goal is to understand and predict LMP spikes for ISO-NE data in real time. By comparing the observation with the prediction, we can detect anomalous price spikes. We started by analyzing ISO-NE dataset that contains LMP, demand, reserve, schedule, and binding constraint data. This dataset is acquired from ISO-NE website and stored in a MySQL database. In particular, the five minute LMP table contains 12,255,233 data recorded from 2020-06-08 17:10:00 to 2020-07-30 16:50:00 at 1209 locations. The basic statistics of the LMP and its energy component, congestion component, and loss component are given in Table 4.24.

Note that the highest LMP is at $673.26 and the lowest LMP is at $-150.00. Figure 4.61 shows the LMP data at the locations with the highest and the lowest prices.

From Figure 4.61, it seems that LMP reaches the lowest price at $-150.00 several times as if the price is capped below. A zoom-in plot for one example of these times is provided in Figure 4.62. Note there is a mirror image between energy and congestion costs so as to

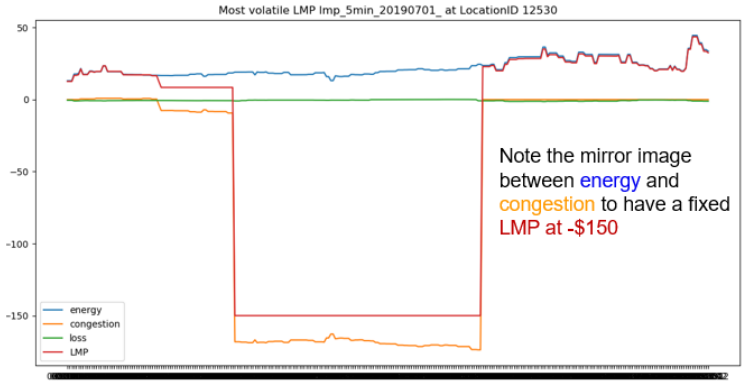


Figure 4.62: Price cap by congestion cost.

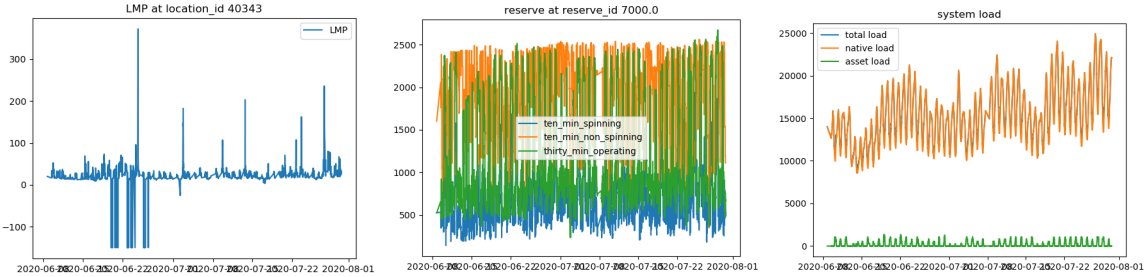


Figure 4.63: Lowest LMP, the reserve, and the system load.

have the total LMP capped at $-150.

This observation reminds us Rule 4 in the price spike model, cited as: ***Rule 4:*** *The regulatory setting for pricing reserve purchases determines the shape of price spikes.* If the rule is sharp, for example, pay up to $5000 if reserves fall below 5% and pay nothing if reserves are above 5%, then the price spike will jump sharply, with a high value and a short duration. On the other hand, if the rule setting increases gradually, then the price spike will be lower and broader. This implies a possible connection between the price cap and the reserve prices.

To explore the relation of reserve price and LMP, we plotted the lowest LMP and the reserve data side by side in Figure 4.63. The reserve data has a clear cycling pattern in all categories (10-minute spinning, 10-minute non-spinning, and 30-minute operating) that is not observed in the LMP data. Similar observation can be made for the system load data.

Hourly LMP Data

In contrast to the five-minute LMP data, the hourly LMP data has the day-ahead forecast. This is because hourly LMP is less volatile than the five-minute LMP, which makes prediction one day ahead possible. For example, Figure 4.64 shows the real-time hourly LMP and the day-ahead hourly LMP at the highest price and the lowest price. Note that while the real-

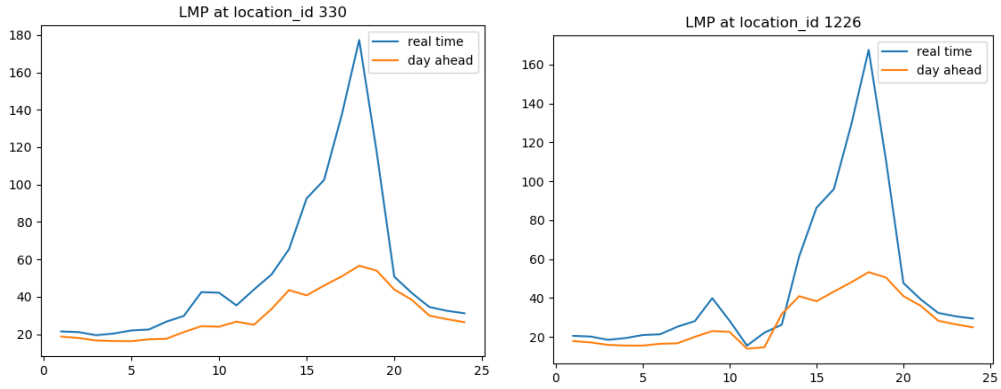


Figure 4.64: Hourly LMP with the highest (left) and the lowest (right) price.

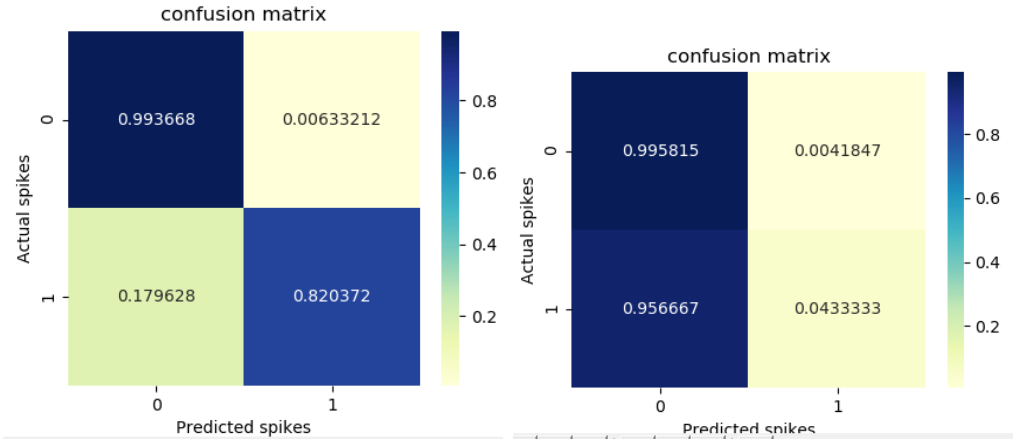


Figure 4.65: Confusion matrix for the logistic regression model in predicting spikes at $100 (left) and $150 (right) in hourly LMP.

time data is more volatile, its upward and downward trends are captured by the day-ahead data. This implies that day-ahead data can be used to predict the spikes in the real-time data.

We started with a simple logistics regression (LR) model with day-ahead LMP as the only feature and the real-time LMP spikes as the target. The spike threshold was set to be $100. For this test, we used 24-hour data from 1205 locations with 28,920 observations. By splitting the dataset into 75% and 25% for training and testing, respectively, the logistic regression model achieves 82% accuracy in prediction shown in Figure 4.65. When the spike threshold is set at $150, however, the prediction accuracy drops to 4%, due to fewer spikes.

We then built two more classifiers, gradient boosting (GB) and random forest (RF), to compare with LR results. Note that gradient boosting and random forest outperform LR significantly, achieving 69.7% and 70.3% prediction accuracy, respectively, as shown in Table

LR	Predicted No	Predicted Yes	**RF**	Predicted No	Predicted Yes	**GB**	Predicted No	Predicted Yes
Actual No	0.995	0.005	Actual No	0.995	0.005	Actual No	0.995	0.005
Actual Yes	0.96	0.04	Actual Yes	0.297	0.703	Actual Yes	0.303	0.697

Table 4.25: Confusion matrices for logistic regression (LR), gradient boosting (GB), and random forest (RF) for LMP spike threshold at $150.

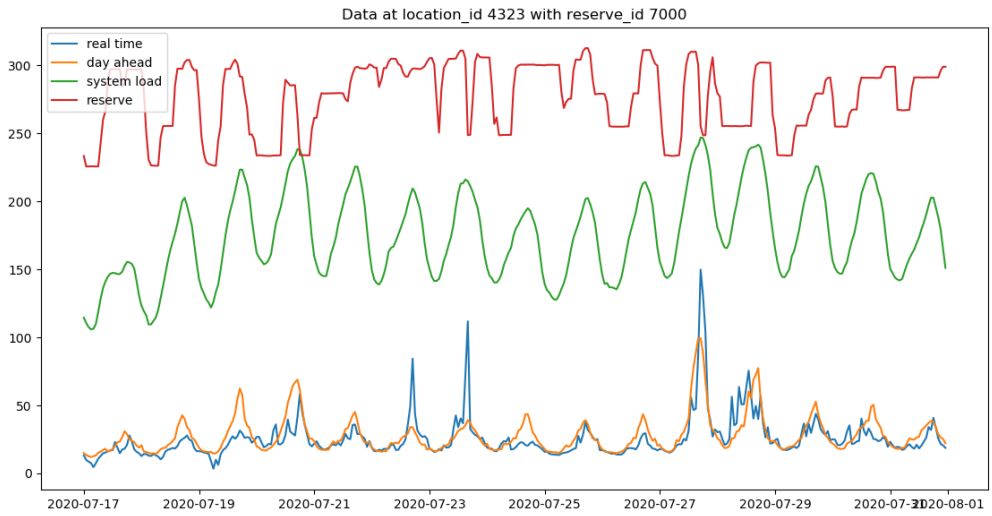


Figure 4.66: System load, reserve, real-time hourly LMP, and day-ahead hourly LMP.

4.25.

While these results are encouraging, the models are trained on the 24-hour dataset, which is very limited. When we use the three models to predict a different dataset, namely, a month hourly data at the location with the highest LMP, all three models miss all the spikes.

Including load and reserve data

Using solely hourly LMP data proved to be insufficient to predict spikes. We then included system load data and reserve data to identify spikes caused by reserve shortage or load surge. These new features are plotted in Figure 4.66 to show their temporal correlation.

Test results show new features improve the performance of all three models significantly. In particular, the logistic regression, the random forest, and the gradient boosting achieve 30%, 61%, and 99% prediction accuracy, respectively. Table 4.26 shows the confusion matrices for all three models.

To understand the contribution of the two new features, system load and reserve, we repeated the experiment by adding only the load or only the reserve. By adding load only, we have almost the same performance for all three models (<1% difference) as we did when

LR	Predicted No	Predicted Yes	**RF**	Predicted No	Predicted Yes	**GB**	Predicted No	Predicted Yes
Actual No	0.997	0.003	Actual No	1	0	Actual No	0.997	0.0003
Actual Yes	0.70	0.30	Actual Yes	0.39	0.61	Actual Yes	0.009	0.991

Table 4.26: Confusion matrices for the three models when system load and reserve data are included.

LR	Predicted No	Predicted Yes	**RF**	Predicted No	Predicted Yes	**GB**	Predicted No	Predicted Yes
Actual No	0.975	0.025	Actual No	0.98	0.02	Actual No	0.97	0.03
Actual Yes	0.42	0.58	Actual Yes	0.52	0.48	Actual Yes	0.39	0.61

Table 4.27: Confusion matrices of three models for multi-year data from 2017 to 2020.

we used both load and reserve. Therefore, reserve is not necessary when load is included to achieve the performance in Table 4.26. On the other hand, by adding the reserve only, we see worse performance for all three models (e.g., gradient boosting spike prediction drops to 90% from 99%). In conclusion, the system load contributes more than the reserve in achieving performance in Table 4.26.

ISO-NE Web Services

Previous price spike models were only trained for one-month of data at one location of ISO New England. To generalize the model, we need more data spreading over multi-year and multi-location in ISO-NE. To this end, we resort to the web services provided by ISO-NE [97].

Given the location ID, we obtained LMP, system load, and reserve data for the entire 2017, 2018, 2019, and partial 2020 (up to August, 27) dataset with 32,037 observations. For this dataset, the performance of the three models is given in Table 4.27. In this case, the gradient boosting model achieves the best performance with prediction accuracy at 61% and the random forest performs the worst at 48%.

Hourly LMP vs. Five-Minute LMP

The ultimate goal for price spike detection is to provide additional information for the five minute anomaly detection. Thus it is critical to study if the hourly LMP can capture the trend in the five-minute LMP. Figure 4.67 shows an example of the LMP data for one day. Here, the five-minute data follows the trend of the hourly data, but more volatile.

The same observation can be made for the one-week data as shown in Figure 4.68. In this case, the spikes of five-minute data are more significant in magnitude than those of hourly data.

A zoom-in plot of 10 days in Figure 4.69 shows, with a threshold of $150, the hourly data miss three out of four spikes in five-minute data. This implies that using models trained on

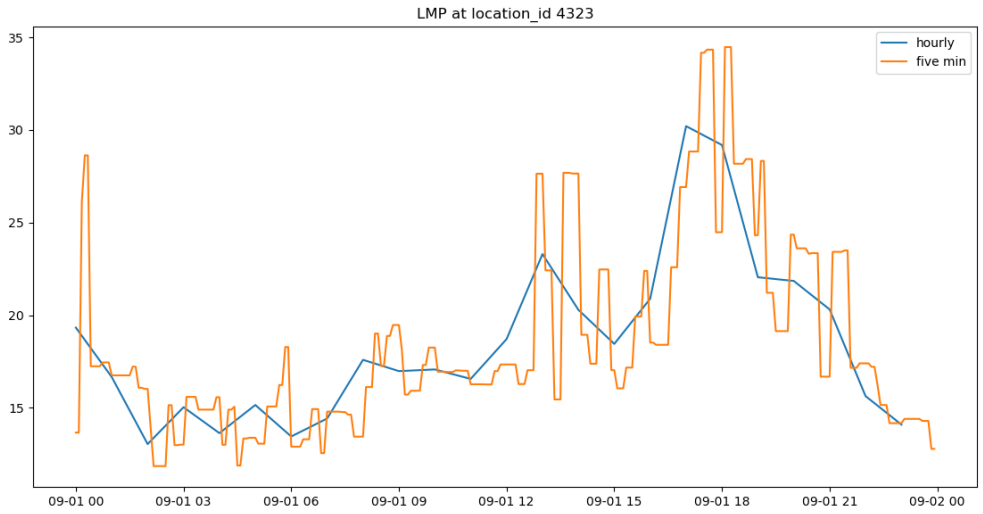


Figure 4.67: Hourly LMP and five-minute LMP in one day.

Spike Threshold	Hourly LMP	Max 5 -min LMP	Overlap	Overlap/Hourly	Overlap/Max 5-min
\$150	7	20	6	6/7=85.7%	6/20=30%
\$100	32	52	28	28/32=87.5%	28/52=53.8%
\$50	156	299	148	148/156=94.9%	148/299=49.5%

Table 4.28: Spike overlap for hourly and five-minute data at different threshold.

hourly data to predict spikes in five-minute data will have a compromised detection accuracy.

To put this into perspective, consider the number of overlapping spikes in hourly and five-minute data shown in Table 4.28. For spike threshold at \$150, six spikes in hourly data overlap with five-minute data with 20 spikes, with a 30% overlapping rate. On the other hand, the overlapping rate for the hourly data is six out of seven with 85.7%. Note that with a lower threshold for spikes, the overlap rate improves in both hourly and five-minute data. This is because there are more spikes observed.

Feature Selection and Hyper-Parameter Tuning

The importance of features in all three machine learning models were studied for feature selection. To recap, we used seven features to predict spikes in real-time hourly LMP data: Day-Ahead Hourly LMP, System Load, Native Load, Ard Demand, 10-min spin reserve, total 10-min reserve, and reserve. Table 4.29 shows the importance of these features for the three machine learning models. We see that day-ahead hourly LMP carries a lot of weight, in particular, 75% and 43% for the gradient boosting and the random forest model, respectively. Note that the logistic regression model in Scikit-Learn does not provide feature

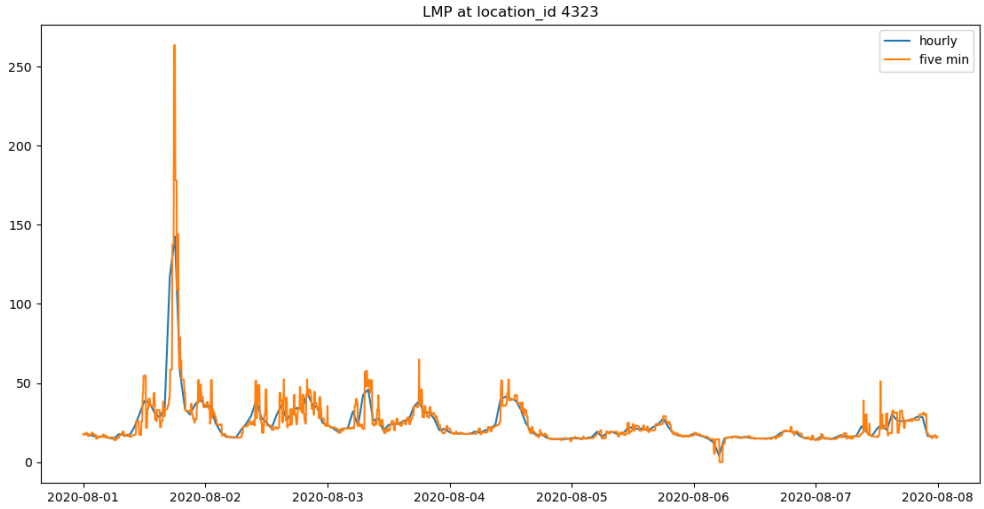


Figure 4.68: Hourly LMP and five-minute LMP in a week.

Feature	**Gradient Boosting**	**Random Forest**
Day-ahead Hourly LMP	0.75	0.43
Load	0.06	0.14
Native Load	0.09	0.28
Ard Demand	0.04	0.19
10-min Spin Reserve	0	0
Total 10-min reserve	0	0
Reserve	0.06	0.13

Table 4.29: Feature importance of the machine learning models.

importance, so the importance weight in for logistic regression is not available in Table 4.29.

Next, we fine tuned parameters for the gradient boosting model, which outperforms the other two models. There are a dozen hyperparameters in gradient boosting, of which learning rate and the number of trees are the most important [98]. The learning rate controls the magnitude of changes in the estimate of gradient boosting, while the number of trees determines the complexity of the model. The default values for the gradient boosting model in Scikit-Learn are learning_rate = 0.1 and n_trees = 100. We conducted the basic grid search for these two hyperparameters. By ranging from defaultValue/10 to defaultValue*10, we have a grid of parameters from learning_rate = [0.01, 1] and n_trees = [10, 1000].

The target metric is the prediction accuracy (e.g., when GB model predicts a spike, it is actually a spike). Figure 4.70 (Left) shows the prediction accuracy for each pair of learning rate and number of trees. The best performance at 66% is achieved at a set of parameters

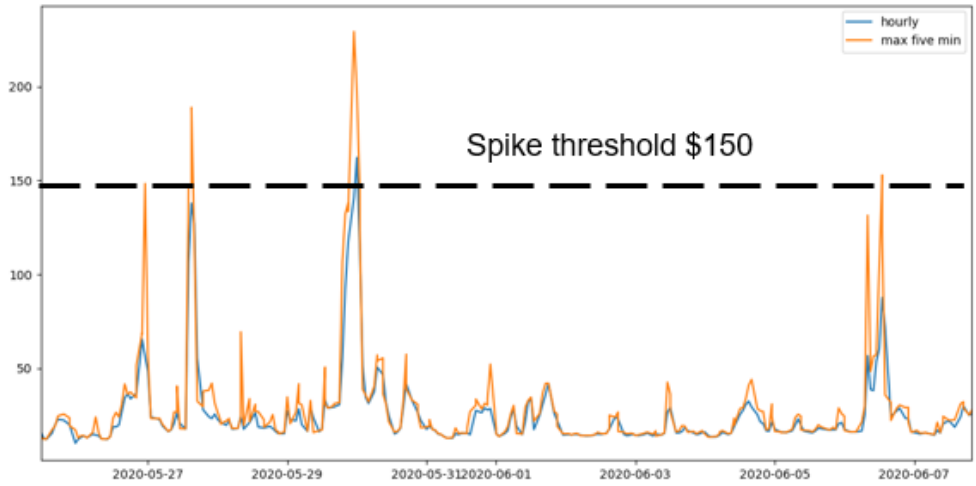


Figure 4.69: Price spikes in hourly and five-minute LMP.

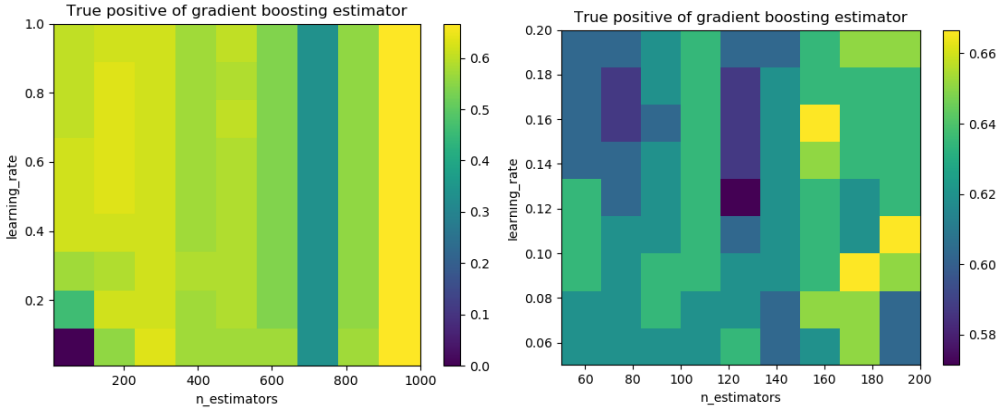


Figure 4.70: Prediction accuracy of gradient boosting model by varying the learning rate and the number of trees.

in coarse ranges, where we fine tuned in small ranges, namely, learning_rate = [0.05, 0.2] and the n_trees = [50, 200]. Figure 4.70 (Right) shows the target metric over this set of parameters. In particular, learning_rate = 0.16 and n_trees = 150 result in the 66% accuracy.

Multi-Zone Data

We then extended the research to all dispatch zones of the ISO-NE dataset. ISO-NE territory is divided into eight different zones: Maine, New Hampshire, Vermont, Connecticut, Rhode Island, Southeastern Massachusetts (MA), Western/Central MA, and Northeastern MA shown in Figure 14.9.

We trained all three models, logistic regression, random forest, and gradient boosting

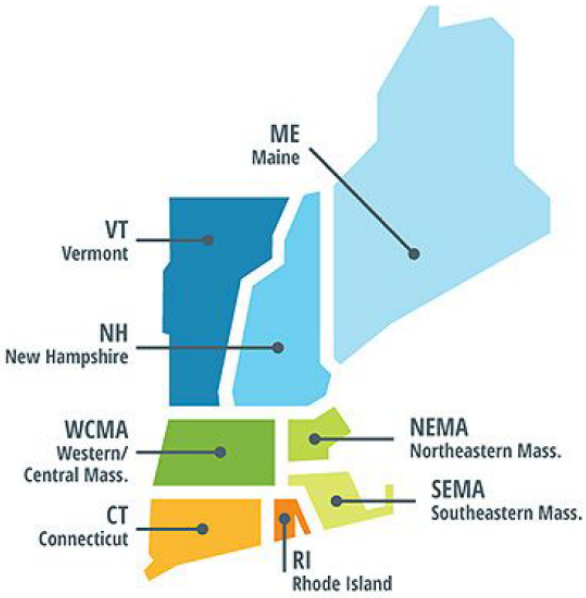


Figure 4.71: Eight zones of ISO-NE [2].

Zone	Logistic Regression	Random Forest	Gradient Boosting
Maine	0.997	1	0.993
New Hampshire	0.998	1	0.996
Vermont	0.998	1	0.995
Connecticut	0.997	1	0.995
Rhode Island	0.997	1	0.996
Southeastern MA	0.995	1	0.993
Western/Central MA	0.998	1	0.996
Northeastern MA	0.994	1	0.993

Table 4.30: True negative rate of machine learning models.

over eight zones. All three models achieve high performance in true negative rate (>99%) consistently over all eight zones, see Table 4.30. On the other hand, they achieve around 60% accuracy for the true positive rate; see Table 4.31.

Weather Data

We next included weather data from the National Oceanic and Atmospheric Administration [99]. In particular, we obtained dry bulb temperature, wet bulb temperature, and humidity. Furthermore, we engineered a few features with respect to time, for example, if the event happened in weekdays or weekends, in working hours between 9:00 to 17:00. With the new weather data and the timing features, we conducted a correlation study with the real-time LMP. Table 4.32 shows the correlation between real-time LMP and the data points. Note

Zone	Logistic Regression	Random Forest	Gradient Boosting
Maine	0.51	0.18	0.47
New Hampshire	0.58	0	0.54
Vermont	0.53	0.25	0.61
Connecticut	0.57	0	0.61
Rhode Island	0.58	0	0.56
Southeastern MA	0.50	0	0.61
Western/Central MA	0.60	0	0.56
Northeastern MA	0.63	0	0.53

Table 4.31: True positive rate of machine learning models.

Feature ID	Feature Name	Correlation
0	lmp_rt	1
1	load_da	0.280402
2	Imp_da 2	0.652581
3	NativeLoad	0.37233
4	ArdDemand	-0.141967
5	reserve	-0.0315462
6	humidity	-0.0941187
7	drybulb	-0.266672
8	wetbulb	-0.274475

Table 4.32: Correlation between real-time hourly LMP and the data points.

that the temperature data is negatively correlated with real-time LMP, which seems a bit surprising at first glance. To understand this, we plotted the correlation between dry bulb temperature and the real-time LMP in Figure 4.72. Note that the temperature is negatively correlated with LMP from November to April and it is positively correlated with LMP from May to October. To gain a full picture, we computed the correlation between LMP and all data points over months; see Table 4.33.

In addition, we performed the seasonality decomposition of the day-ahead load, day-ahead LMP, and real-time LMP, with the results shown in Figure 4.73. Now with these additional features in trend, residual, and seasonal data, and the timing features discussed earlier, we obtained the prediction results shown in Table 4.34.

Voting Machine

We next explored the ensemble of the three tested machine learning models using a voting machine. The basic idea of a voting machine is to combine a diverse set of models with majority to vote for a stronger prediction model; see Figure 4.74 for an illustration.

To see how the voting machine works, we plot the false positive of three models, logistic

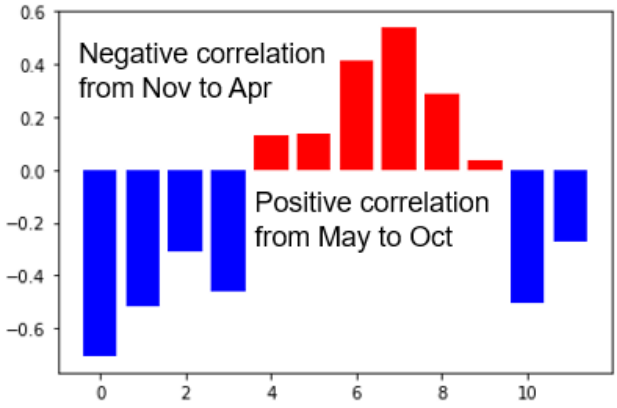


Figure 4.72: Correlation between dry bulb temperature and real-time LMP.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0	1	1	1	1	1	1	1	1	1	1	1	1
1	0.523	0.3836	0.2717	0.3713	0.2967	0.2475	0.5678	0.6172	0.2366	0.2287	0.4733	0.3719
2	0.8667	0.8209	0.4844	0.577	0.3858	0.292	0.5757	0.7368	0.2455	0.3922	0.7178	0.7239
3	0.6254	0.4904	0.3968	0.5476	0.4261	0.3808	0.6418	0.6549	0.3157	0.3357	0.573	0.5221
4	-0.217	-0.1897	-0.2485	-0.2756	-0.3168	-0.1416	-0.2594	-0.2627	-0.06059	-0.1932	-0.2711	-0.112
5	-0.0859	0.3666	0.1027	-0.06049	0.1146	-0.2668	-0.09795	-0.1354	-0.07824	-0.112	0.4115	-0.3602
6	-0.3055	-0.1255	-0.06861	0.3052	0.03922	0.1212	-0.04574	-0.3781	-0.1122	0.07885	-0.04861	-0.1212
7	-0.705	-0.5137	-0.3078	-0.4572	0.1324	0.1384	0.4137	0.5383	0.2879	0.03704	-0.5046	-0.274
8	-0.688	-0.5003	-0.291	-0.3025	0.2268	0.2538	0.4037	0.4073	0.2165	0.05947	-0.44	-0.2542

Table 4.33: Correlation between LMP and data points over 12 months. The indices of the data points are given in Figure 25. The horizontal axis denotes month data.

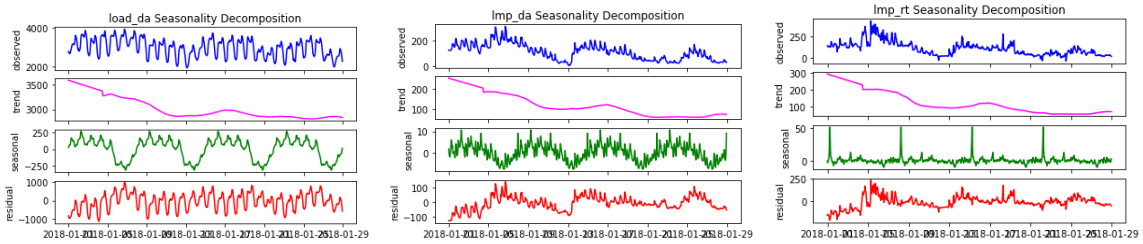


Figure 4.73: Seasonality decomposition of the day-ahead load data (left), the day-ahead LMP data (middle) and the real-time LMP data (right).

regression, random forest, and extreme gradient boosting (the enhanced version of GB) in Figure 4.75. Note that by combining the votes from three models, the voting machine has better confidence in the prediction results. Similarly, the false negative votes from three individual models and the aggregated votes are shown in Figure 4.76.

Nevertheless, the performance of voting machine does not outperform the individual models. Recall that price spikes are defined as price values that surpass a threshold (e.g.>$1000) The missed detections happen mostly on the spikes at marginal distance to the spike threshold. This indicates threshold selection is critical to the detection accuracy.

LR	Predicted No	Predicted Yes	**RF**	Predicted No	Predicted Yes	**GB**	Predicted No	Predicted Yes
Actual No	2112	10	Actual No	2122	0	Actual No	2113	7
Actual Yes	28	36(34)	Actual Yes	34	30(18)	Actual Yes	22	42(45)

Table 4.34: Confusion matrices for the machine learning models with new data points.

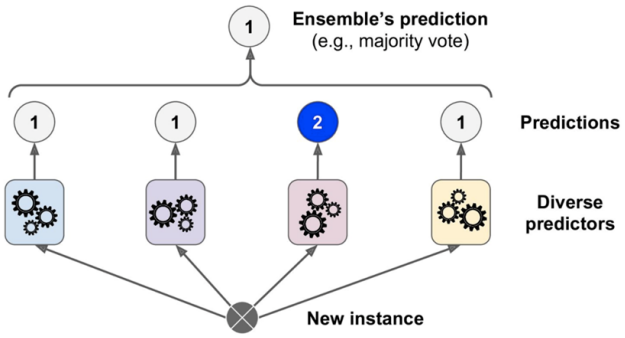


Figure 4.74: Voting machine prediction [3].

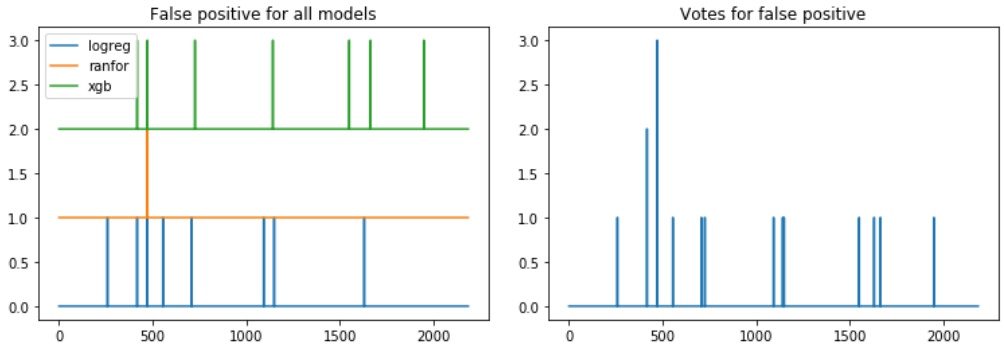


Figure 4.75: False positive votes from individual models (left) and the aggregated votes (right).

Spike Thresholds

We next varied the threshold of the price spikes and examined the robustness of all models with respect to the thresholds. The reason for this sensitive test is that the user (e.g., operators at ISO-NE) may choose their own spike threshold. A model whose performance is consistent over price thresholds will be more valuable.

Figure 4.77 shows that the number of spikes decreases as the threshold increases. For each of the thresholds, we repeated the same exercise for all models. Figure 4.78 shows that the false alarm rate drops first when the threshold increases and hovers around a fixed value

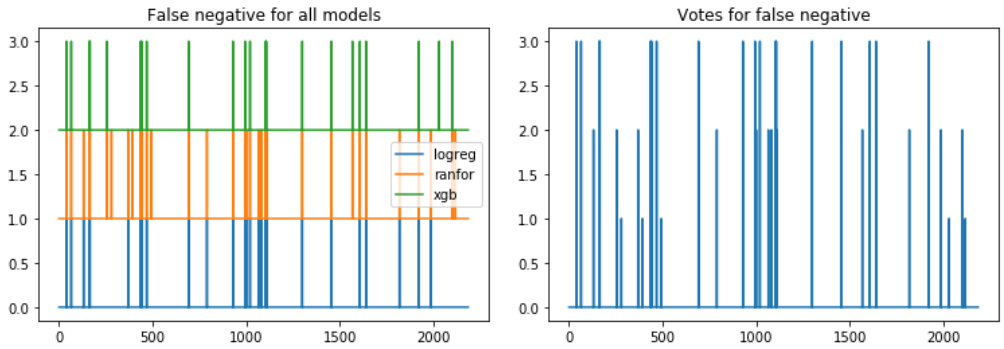


Figure 4.76: False negative votes from individual models (left) and the aggregated votes (right).

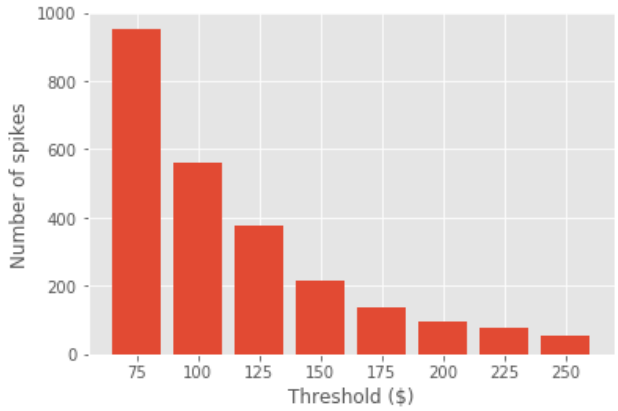


Figure 4.77: Number of spikes for varying thresholds.

after $150. The detailed numbers for the false alarm rate are given in Table 4.35.

Figure 4.79 shows the accuracy of all machine learning models. Note that the performance of logistic regression and random forest drops significantly when the threshold increases. The voting machine, however, follows the same trend when it is fed with results from these models. On the other hand, the performance of the gradient boosting model and its enhanced version, XGB, degrades gracefully as the threshold increases. In particular, XGB manages to achieve 60% prediction accuracy when other models suffer from the performance loss at high values of price thresholds, as shown in Table 4.36.

4.6.5 Conclusion

In this section, we studied the price spike patterns and mechanisms. We then designed algorithms for spike analysis and prediction and tested them on ISO-NE data. We developed data acquisition pipelines from several sources including ISO-NE and NOAA. By taking

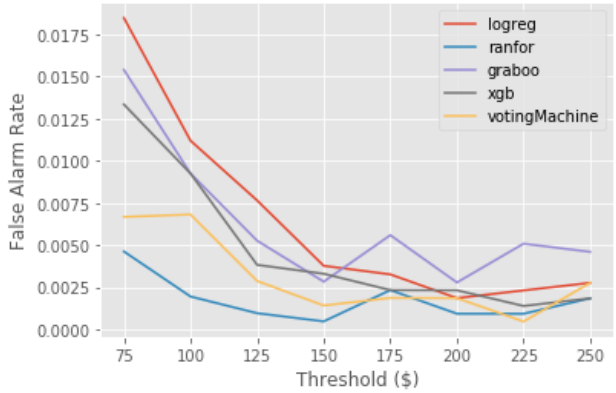


Figure 4.78: False alarm rate for all machine learning models.

Threshold	LR	RF	GB	XGB	VotingMachine
75	0.0185	0.0046	0.0154	0.0133	0.0067
100	0.0112	0.0019	0.0093	0.0093	0.0068
125	0.0077	0.001	0.0053	0.0038	0.0029
150	0.0038	0.0005	0.0028	0.0033	0.0014
175	0.0033	0.0023	0.0056	0.0023	0.0019
200	0.0019	0.0009	0.0028	0.0023	0.0019
225	0.0023	0.0009	0.0051	0.0014	0.0005
250	0.0028	0.0018	0.0046	0.0018	0.0028

Table 4.35: False alarm rate in numbers.

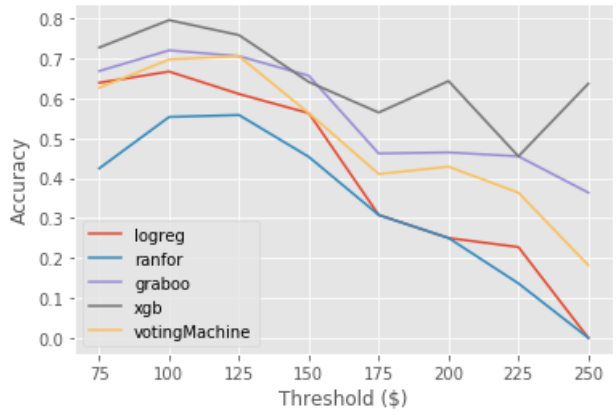


Figure 4.79: Prediction accuracy for all machine learning models.

Threshold	LR	RF	GB	XGB	VotingMachine
75	0.6387	0.4244	0.6681	0.7269	0.6261
100	0.6667	0.553	0.7197	0.7955	9.697
125	9.6105	0.5579	0.7053	0.7579	0.7053
150	0.5625	0.4531	0.6562	0.6406	0.5625
175	0.3077	0.3077	0.4615	0.5641	0.4103
200	0.25	0.25	0.4643	0.6429	0.4286
225	0.2273	0.1364	0.4545	0.4545	0.3636
250	0	0	0.3636	0.6364	0.1818

Table 4.36: Prediction accuracy of all models in numbers. Note that XGB stays around 60% when other models have suffered performance loss at higher values of price thresholds.

advantages of the web services provided by ISO-NE, the developed pipeline enabled us to pull, process, and store data automatically. We compared multi-year and multi-location datasets. We engineered new features based on the event time and the seasonality of time series. In addition, we employed state-of-the-art machine learning models, including gradient boosting and its enhanced version, to predict LMP spikes. By limiting to realistic day-ahead data, our models achieved less than 1% false alarm rate and 66% prediction rate. Among these models, the extreme gradient boosting is robust with respect to the spike thresholds, which makes it the best option for real world implementation.

4.7 Conclusion

In this chapter, we elaborated the three categories of anomaly detection algorithms: the point-wise anomaly detection, the locational anomaly detection and the price spike anomaly detection. For each algorithm class, we explained the detection mechanism, the parameter optimization, the threshold selection and the testing and evaluation results that lead to the achievement of project milestones. Both real-world datasets (PJM and ISO-NE) and simulation datasets were used for signature study and performance evaluation. Collectively, these algorithms form the WISP data-driven detection core.

Chapter 5

WISP Algorithms: Electricity Market Vulnerability Analysis

5.1 Introduction

The interconnected communication network presents the modern power system operation unprecedented threats from cyberattacks. For instance, in December 2015, the information system for three distribution centers in Ukraine was compromised, and 30 substations were switched offline [100]. In March 2019, a denial of service attack happened at a western utility in the U.S. disconnecting the communication between operators and remote generation sites for a minute [101]. Those real-life events demonstrate that cyber intrusions are capable of penetrating the communication systems in power grid operation.

The U.S. power market clears hundreds of Gigawatt loads every hour, where electricity is produced reliably and economically. Therefore, malicious communication breaches into market operations could induce catastrophic consequences on fair financial settlements and reliable transmission services. Followed by the initial discussion of market-targeted cyberattacks presented in [32], there is abundant literature discussing various cyberattacks on power market operations

5.1.1 Literature Review

Three main directions of market-targeted cyberattacks are summarized as follows:

(1) The development of new attack strategies

In the first category, state estimation (SE) is the most popular intrusion path. In [102], a robust false-data injection attack (FDIA) on SE is designed to create a financial bias on market settlements along with bogus bids. In [103], an undetectable parameter attack on the system model is designed for financial profit in market operations. In [104], a topology attack is combined with a FDIA to lead customers to pay a higher bill through undetectable

price deviations. In [105], three new topology attacks on SE are developed to mislead both economic dispatch and reliable operation. Next, [106] realizes that the grid topology is too extensive to be known by attackers and proposes a new profitable attack method without prior information on grid topology. Similarly, imperfect topology information is dealt with via robust optimization and stochastic programming in [107] and [108]. Various new attack paths and scenarios on market operation are identified: a transmission line rating attack [109], a ramping constraints attack [110], and very short-term load forecasting [111].

(2) The development of new detection schemes

For the second category of market-targeted cyberattacks, developing new detection schemes, detecting cyberattacks on market operations mainly focuses on SE level protections. In [112], a least-budget defense algorithm is proposed to secure pre-selected sensors, leading to the failure of bad data detection attacks. Additionally, [113] and [114] focus on enhancing the bad data detection algorithm itself because random noise has consistent statistic distributions, but a FDIA changes the pattern.

(3) The development of the sensitivity studies on cyberattacks

In the last category of market-targeted cyberattacks, investigating the sensitivity of cyberattacks, sensitivities of SE manipulation on market-clearing results are fully investigated. In [68] and [115], the sensitivity of locational marginal prices (LMPs) to bad meter data is formulated, and buses with higher sensitivity are prone to being attack targets. In [116], the mathematic representation for the sensitivity of profitability to topology data is investigated. In [117], the sensitivity of renewable generation curtailments to profitability is formed. Although the curtailments in [117] are described as a strategy, the malicious attack could lead to the same results.

5.1.2 Significance of Cyber-Vulnerability Analysis

As presented in the previous subsection, various market cyberattacks and their corresponding defense strategies have been identified and demonstrated. They generally focus on elaborating the attack paths or specific strategies. However, from the market operators' viewpoint, no matter where the attack path lays, whether in the state estimation or the market gateway, the potential targets for a market operation are as follows: unit bids, demand management, generation capacities, line ratings, and congestion patterns. Therefore, it is important for the market operator to identify the vulnerability among all those attack paths. To the best of our knowledge, no previous research has developed a comprehensive analysis model regarding the vulnerability of the electricity market model involving all potential attack objectives and targets. Therefore, this work first provides an impact analysis model that emulates the market-clearing under various cyberattacks, and then introduces a set of algorithms to identify the vulnerability from different aspects. The significance of this cyber-vulnerability analysis are as follows:

1. A comprehensive cyber-vulnerability analysis (CVA) model is provided, in which mar-

ket data from all sources is assumed to be susceptible to attacks, including line ratings, congestion patterns, generation capacity withholds, market-interface, etc. Namely, all parameters in the ISO's market model are assumed to be attackable. Next, various attack objectives are categorized and considered. The market operator can apply the model to perform impact analysis on market cyberattacks.

2. Four specific impact analysis algorithms are provided to identify the vulnerability of power market parameters comprehensively. The CVA algorithms target at four vital aspects regarding the vulnerability of power market parameters: (1) The vulnerability in terms of the possibility: which attack paths are the most likely of being attacked? (2) The vulnerability in terms of the severity: which attack paths have the most impact on the market operation? (3) The vulnerability in terms of the load level: which load level is more likely for attacks to happen? (4) The vulnerability in terms of the defense strategies: how the defense actions impact the effectiveness of market cyberattacks?

5.2 Cyber-vulnerability Analysis

The previous subsection discusses the research gap in power market cybersecurity literature. This report will detail a comprehensive CVA platform for delivering a detailed analysis from four aspects: highly probable cyberattack targets, devastating attack targets, risky load levels, and mitigation ability under different degrees of defense. Users can simulate the interactions between attackers and defending operators under different attack events, and the corresponding market settlements can also be obtained.

5.2.1 Cyber-Vulnerability Analysis Model

The analytic model provides a flexible platform to emulate different attack strategies and defense degrees under various assumptions. The details of the vulnerability analysis algorithms are discussed in the next section. This section presents the structure of the CVA model.

The CVA model simulates an attacker and a defending market operator. The attacker wants to optimize its objective (e.g., LMP manipulations), then it anticipates the optimal response from the market operator. In this setting, the attack's optimization problem contains a nested optimization task that corresponds to the market operator's optimization problem. The defending ability is modeled for the impact analysis of defense degrees. Therefore, the proposed model is constructed as a bilevel optimization problem. The attacker modifies the parameters that impact the market-clearing result, and the market operator clears the market with defending variables, which in turn affects the attacker's objective. The overall structure of the proposed model for CVA is shown in Figure 5.1.

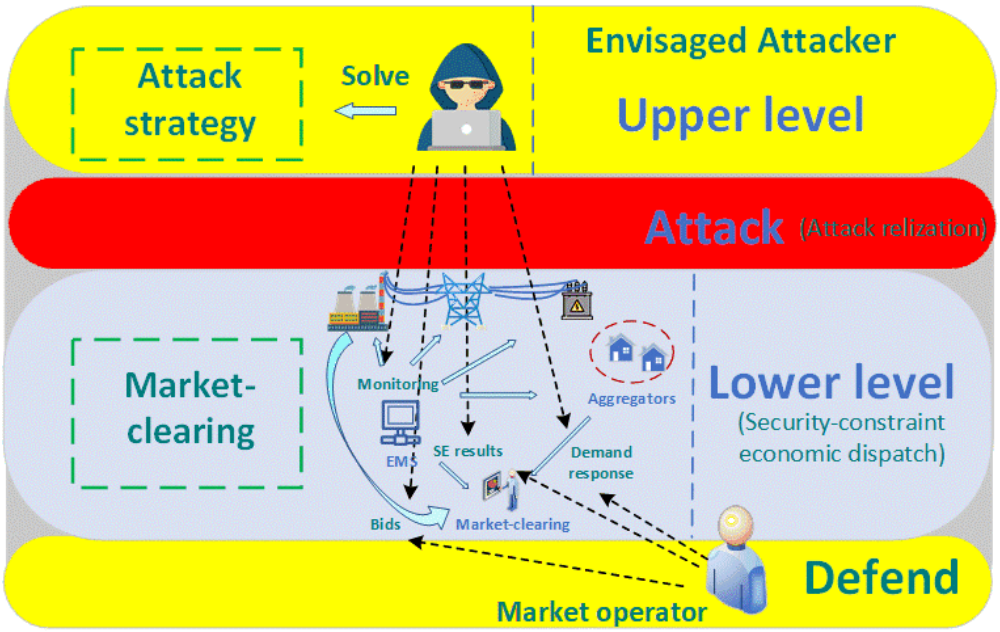


Figure 5.1: CVA model structure

Table 5.1: Potential attack objectives.

Type	Objective	Model
Financial settlements	LMP	LMP_i
	Social-welfare	$\sum C_i(P_i)$
Generation	Generation dispatch	P_i
Transmission	Congestion price	$LMP_i - LMP_j$
	Congestion pattern	L

(1) Upper level model: attacker

Although most of the existing research assumes that attacks on the market are profit-driven, the purpose of cyberattacks on market operation varies from one attacker to another. Generally, potential objectives for power market cyberattacks can be categorized into three types: (1) financial settlements, (2) generation dispatches, and (3) transmission congestions. Therefore, the proposed model considers different attack objectives from each of the above categories, as shown in Table 5.1 to provide a general attack evaluation. The objective of the upper level model can be selected from Table 5.1 based on different analysis purposes.

The upper level of the analysis model incorporates all potential attack targets in market operation. When the market operator solves a real-time economic dispatch problem, data from multiple sources are used, including: (1) short term load forecasts and demand management from energy management systems (EMSs); (2) bidding price and generator

capacity from market gateway; and (3) congestion patterns and line ratings from EMSs. Therefore, to conduct a comprehensive analysis, all of the above data sources are assumed to be susceptible to attacks, as shown in Equation 5.1-5.7. Although some parameters may not be easily compromised unless the cyber threats are from insiders, the proposed CVA model considers comprehensive scenarios to provide a general analytic platform for market operators to identify possible cyber vulnerabilities. Specific constraints and variables can be simplified or removed if decision makers consider these parameters perfectly secure. The maximum amount of those attacks is constrained by the penetration level value q and the targets' original value.

$$-q_i^b c_i B_i^b \leq M_i^b \leq q_i^b c_i B_i^b, \forall i \in NG \tag{5.1}$$

$$-q_i^d d_i B_i^d \leq M_i^d \leq q_i^d d_i B_i^d, \forall i \in L \tag{5.2}$$

$$-q_i^p P_i^{max} B_i^p \leq M_i^p \leq q_i^p P_i^{max} B_i^p, \forall i \in NG \tag{5.3}$$

$$-q_l^{L+} L_l^{max} B_l^{L+} \leq M_l^{L+} \leq q_l^{L+} L_l^{max} B_l^{L+}, \forall i \in L_{\delta_l^+} \tag{5.4}$$

$$-q_l^{L-} L_l^{min} B_l^{L-} \leq M_l^{L-} \leq q_l^{L-} L_l^{min} B_l^{L-}, \forall i \in L_{\delta_l^-} \tag{5.5}$$

$$\delta_l^- + \delta_l^+ \leq 1, \forall l \in L \tag{5.6}$$

$$\sum_l (\delta_l^- + \delta_l^+ + B_l^{L-} + B_l^{L+}) + \sum_i (B_i^b + B_i^p + B_i^d) \leq A_{ak} \tag{5.7}$$

where:

$\boldsymbol{A_{ak}}$ is Attack degrees;

$\boldsymbol{q_i^b, q_i^d, q_i^p q_i^{L+}, q_i^{L-}}$ are Penetration level of data manipulation in bid of i_{th} unit, load at i_{th} bus, capacity of i_{th} unit, up flow limit of l_{th} branch, and down flow limit of l_{th} branch;

$\boldsymbol{M_i^b, M_i^d, M_i^p M_i^{L+}, M_i^{L-}}$ are Attack value for bid of i_{th} unit, load at i_{th} bus, capacity of i_{th} unit, up flow limit of l_{th} branch, and down flow limit of l_{th} branch;

$\boldsymbol{\delta_l^+, \delta_l^-}$ are Attack decision for congestion status of l_{th} up/down line flow limits;

$\boldsymbol{B_i^b, B_i^d, B_i^p B_i^{L+}, B_i^{L-}}$ are Attack decision for bid of i_{th} unit, load at i_{th} bus, capacity of i_{th} unit, up flow limit of l_{th} branch, and down flow limit of l_{th} branch.

Constraint in Equation 5.6 means that congestion pattern attacks happen either at upper or lower limits because a line flow can either be on the upper or lower limit. The attacker degree is constrained in Equation 5.7, which represents how many targets the attacker can compromise.

(2) Lower level model: market operator

The market operator is placed at a lower level equipping the capability of defending attacks. The traditional economic dispatch model is reformulated as Equation 5.8 with the considered attacks and corresponding defenses. To identify the critical attack path and defense efficiency, the defense degree is constrained in Equation 5.8f, which means how many attacks can be defended. Although defenders always want to defend all possible attacks, there is always a recourse limit such that we have to defend the most threatening attacks based on the defender's choice. It is worth noting that the defender knowns where the attacker attacked in this bilevel formulation. However, the defender analysis is aiming at analyzing the effectiveness of the defense degree. Equations 5.8g and 5.8h indicate that if an attack is identified, then it is totally countered, and Equation 5.8i shows the defense is only placed where the attack happens.

$$\text{minimize} \quad \sum_i (C_i + M_i^b - N_i^b) P_i \tag{5.8a}$$

$$\text{subject to} \quad \sum_i P_i - \sum_i (d_i + M_i^d - N_i^d) = 0 \tag{5.8b}$$

$$0 \le P_i \le P_i^{max} + M_i^p - N_i^p, \forall i \in NG \tag{5.8c}$$

$$(\delta_l^+ | V_l^{\delta^+}) \sum_i^{N_b} GSF_{l-i}(P_i - (d_i - M_i^d + N_i^d)) \le L_l^{max} + M_l^{L^+} - N_l^{L^+} \tag{5.8d}$$

$$(\delta_l^- | V_l^{\delta^-}) \sum_i^{N_b} GSF_{l-i}(P_i - (d_i - M_i^d + N_i^d)) \le L_l^{min} + M_l^{L^-} - N_l^{L^-} \tag{5.8e}$$

$$\sum_i \sum_j V_i^j + \sum_l \sum_j V_l^j + \sum_l V_l^{\delta^+} + \sum_l V_l^{\delta^-} - A_{df} \le 0 \tag{5.8f}$$

$$N_i^j = V_i^j M_i^j, \forall i \in N^b, \forall j \in \{d, p, b\} \tag{5.8g}$$

$$N_l^j = V_l^j M_l^j, \forall l \in L, \forall j \in \{L^+, L^-\} \tag{5.8h}$$

$$V_i \le B_i, \forall i \in \{d, p, b, L^+, L^-\} \tag{5.8i}$$

where:

$\boldsymbol{C_i}$ is Bidding prices of i_{th} unit;

$\boldsymbol{d_i}$ is Load at bus i;

$\boldsymbol{P^{max}}, \boldsymbol{P^{min}}$ are Up and down generation limits for unit i;

$\boldsymbol{GSF_{l-i}}$ is Generation shift factor which gives the fraction of a change in the injection at bus i that appears on a branch l;

$\boldsymbol{L_l^{max}}, \boldsymbol{L_l^{min}}$ are Up and down transmission capacity for branch l;

$\boldsymbol{A_{df}}$ is Defense degrees;

$\boldsymbol{P_i}$ is Scheduled generation for unit i;

$\boldsymbol{V_l^+, V_l^-}$ are Defense decision for congestion status of l_{th} up/down line flow limits;

$\boldsymbol{V_i^b, V_i^d, V_i^p, V_i^{L^+}, V_i^{L^-}}$ are Defense decision for bid of i_{th} unit, load at ith bus, capacity of i_{th} unit, up flow limit of l_{th} branch, and down flow limit of l_{th} branch;

$\boldsymbol{N_i^b, N_i^d, N_i^p N_i^{L^+}, N_i^{L^-}}$ are Defense value for bid of i_{th} unit, load at i_{th} bus, capacity of i_{th} unit, up flow limit of l_{th} branch, and down flow limit of l_{th} branch.

The proposed model is used to perform the CVA from four different aspects, which will be elaborated in the next section.

5.2.2 Cyber-Vulnerability Analysis Algorithms

Potential attack targets, risky operating conditions, and defense effectiveness are the most vital elements in developing a defense strategy. Therefore, the following four aspects are selected to construct the CVA analysis algorithms.

Algorithm 1: Identifying highly probable attack target

Some parameters are compromised more frequently than other parameters. For example, congestion patterns can be a vital attack route for both LMP manipulation and diminishing social-welfare. As shown in Figure 5.2, protection of the congestion pattern makes those two types of market attackers hard to achieve their desired goal. Therefore, in Algorithm 1, the CVA model is solved iteratively for all interested attack objectives, and the attack route for each attack objective is recorded. The frequently attacked parameters (routes) are identified as vulnerable parameters in terms of the probability of being attacked. Providing protection to the identified parameters diminishes overall attack interest in the market operation. Further, the attacker has different optimal attack routes when they have different attack degrees.

Therefore, market operators can also identify vital attack routes under different attack degrees through Algorithm 1. The detailed procedure of this identification is shown in Algorithm 1.

Algorithm 2: Identifying devasting attack targets

Different from highly probable attack targets (Algorithm 1), devasting attack targets vary from one attack objective to another. The attacks on one parameter could be more effective than the attacks on other parameters for a particular attack objective. As shown in Figure 5.3, modifying load information could be more effective than modifying line rating. Thus, protection of these attack targets largely diminishes the attackers' interests for a specific attack objective. It should be noted that the attack on the congestion pattern is not applicable to this algorithm because the congestion status is a binary variable that does not have a penetration level.

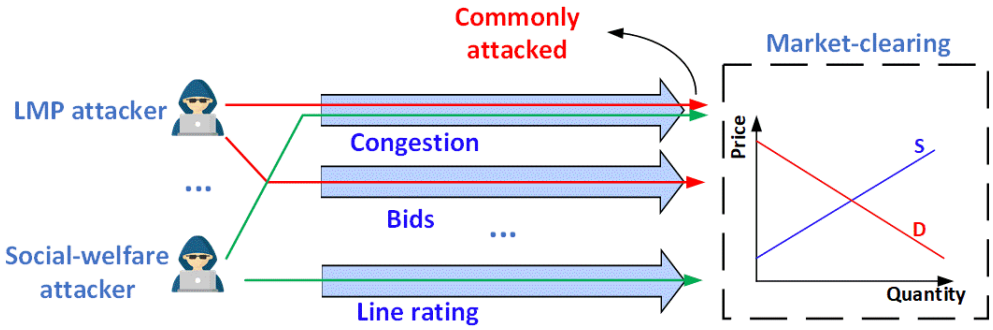


Figure 5.2: Identifying highly probable attack targets

Algorithm 1 Function HPA (market parameters, attack objectives)

Input Real-time market parameters and interested attack objectives
Output Highly probable attack targets

```
for each possible attack degree do
    for each attack objective in Table 5.1 do
        Solving the CVA model (7) - (22)
        Record the attack binary variable B for each target
    end for
    Sum variable B in all attack objectives for each target
end for
Identify targets that have high values of sum(B)
```

Return: the Identified Targets

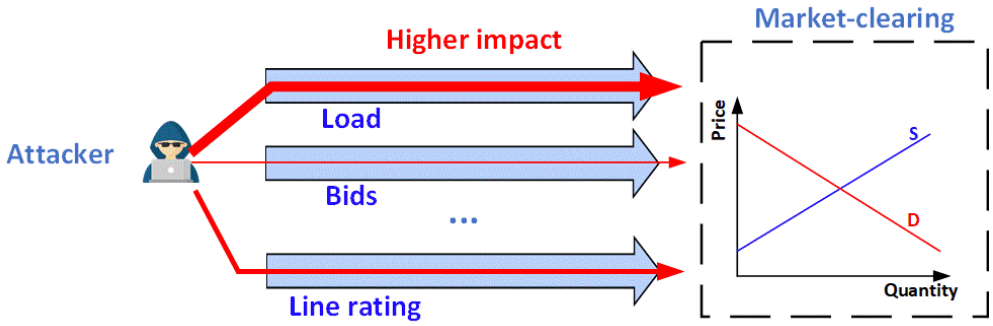


Figure 5.3: Identifying devasting attack targets

Further, LMPs experience step changes regarding some attack routes, such as attacks in load levels, which means the LMP does not change until the modified parameter is large enough. For these attack scenarios, Algorithm 2 can identify the critical attack penetration level that leads to the step change. In Algorithm 2, the CVA model is solved iteratively with a gradual increase of the penetration level Δq under an interested attack objective. The selection of Δq is based on the market operator's need, and the smaller the Δq, the higher the accuracy that can be obtained. The detailed procedure of this identification is shown in Algorithm 2.

Algorithm 2 Function DAT (market parameters, attack objectives)

Input Real-time market parameters and interested attack objectives
Output Devasting attack targets

```
Select interested attack objective from Table 5.1
for each attack target do
    Set attack variables B associated with other attack targets equal to 0
    while penetration level q is less than a threshold do
        Solving the CVA model (7) - (22)
        q = q + Δq
        Record the value of attack objective
    end while
end for
Compare the slope of different attack targets
Identify targets that have steep slopes
```

Return: the Identified Targets

Algorithm 3: Formulating risky load levels

Different load levels result in different market settlements and dispatches. Therefore, the load level is a critical element of a successful cyberattack. As shown in Figure 5.4, the attacker with the same ability could obtain different profits from the market-clearing under different load levels. Therefore, the higher the profitability is, the riskier the load level is. In Algorithm 3, the CVA model is solved iteratively with all interested attack objectives at different load levels. The obtained attack objective values are scaled and summed for each load level. If the value is higher than a threshold, then the load level can be identified as risky. In this study, the same load participation factors are assumed. If the market operator is interested in different load participation factors, the load level and the participation factors are both recorded when solving the CVA model, and the risky load level becomes a risky set containing a load level and load participation factors.

The market operator should take extra caution when the current load level is identified as risky. The detailed procedure of this identification is shown in Algorithm 3.

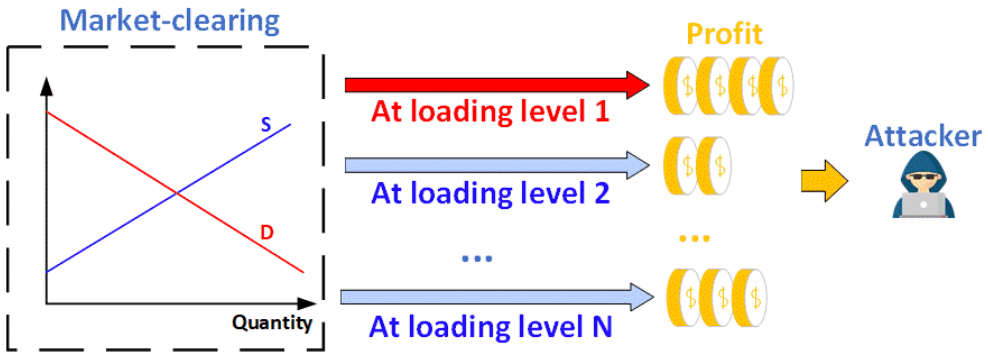


Figure 5.4: Formulating risky load levels

Algorithm 3 Function RLL (market parameters, attack objectives)

Input Real-time market parameters and interested attack objectives
Output Risky load levels

for each load level **do**
  Obtain market-clearing result with/without attacks
  for each interested attack objective **do**
    Solving the CVA model (7) - (22)
    Record the difference between the attacked value and the normal value
  end for
  Sum attack objectives with specified weights $\sum W_i \dot{obj}_i$
end for
Identify load levels that have high weighted values
Return: the Identified Load Levels

Algorithm 4: Investigating the mitigation ability of different defense levels

The goal of Algorithm 4 is to investigate the impact of defense degrees on the effectiveness of the attack. As shown in Figure 5.5, if some of the most effective attack routes are defended by the operator, the attacker might switch to other attack routes. However, those backup attack routes are not as effective. Therefore, investigation of the defense degree to which the attacker may lose interests in attacking is an important aspect of the development of defense strategies. The Algorithm 4 solves the CVA model iteratively with a gradual increase of defense degrees, and the corresponding value of the attack objective is recorded. When the value of the attack objective discourages the attack, the defense degree is identified as the critical defense degree.

The detailed procedure of this identification is shown in Algorithm 4.

The four analysis algorithms described above are demonstrated with examples in the

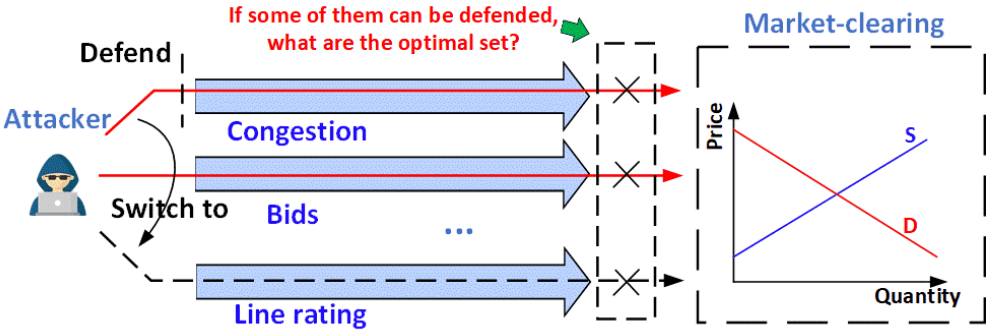


Figure 5.5: The mitigation ability of different defense levels

Algorithm 4 Function DDL (market parameters, attack objectives)

Input Real-time market parameters and interested attack objectives
Output Defense mitigation ability plot

for `each attack objective` **do**
  `Set an interested attack degree` A_{ak} `and set the defense degree` $A_{df} = A_{ak}$
  while defense degree A_{df} is larger than 0 **do**
    `Solving the CVA model (7) - (22)`
    `Record the objective value`
    $A_{df} = A_{df} - 1$
  end while
  `Plot the objective value versus defense degree`
end for
Return: the plots

simulation study. Analysis will be performed using the attack objectives in Table 5.1, but future users can integrate any additional attack objectives in a similar way. The analysis algorithms aim to solve the CVA model iteratively, which could raise a concern on the scalability. Indeed, the number of combinations of attack objectives and attack targets can be astronomical for a real system. However, the potential attack objectives and attack targets could be filtered to a much small portion depending on ISOs or the decision marker's preference. For example, the ISO New England system has 2771 branches but the average active transmission constraints in January 2020, their winter peak month, only have 142 branches [118]. The attacker's ability is also generally a small number because the attacker may not have access to all parameters. Therefore, the number of combinations could be reduced to a small number. Further, the algorithms are for the purpose of analyzing the vulnerability instead of protecting the market operation in real-time. Thus, the analysis

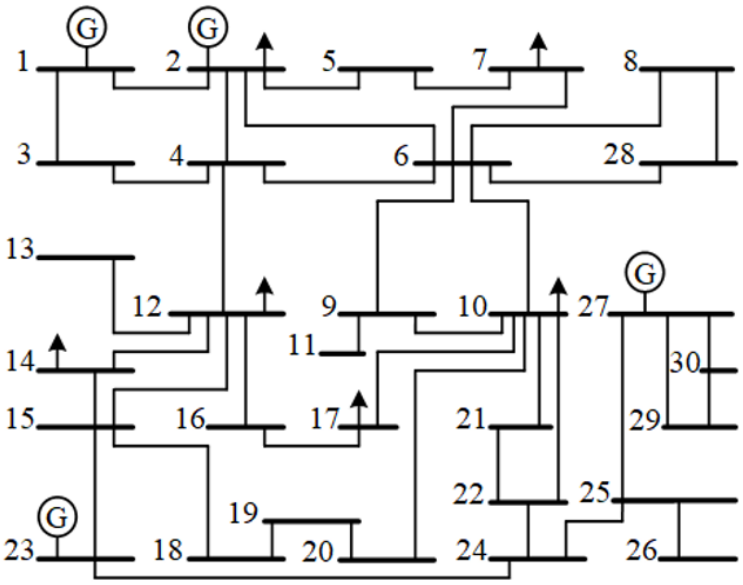


Figure 5.6: One-line diagram of IEEE-30 bus system

could be performed offline and in the cycle of a few weeks (or even months) depending on the market operator's preference. Therefore, the computation is a minor concern for the current vulnerability analysis algorithms.

5.3 Case Study

5.3.1 Test System Description and Simulation Settings

The simulation study was performed on an IEEE 30-bus system. The one-line diagram of the test system is shown in Figure 5.6. Four generators are considered in bus 1, 2, 13, and 27. The detailed system parameters and cost data can be found in [119]. The simulation studies were performed with Matlab 2018 on a PC with Intel i7-8650U processor and 8GB RAM.

5.3.2 Simulation Results and Discussions

Identifying highly probable attack target This study aims to demonstrate Algorithm 1, which identifies highly probable attack targets. The CVA model is solved iteratively for various attack objectives from Table 5.1. The computational time of Algorithm 1 in this study is 70.32 s. Figure 5.7 shows various attacked parameters for each attack objective. The Y-axis shows different objectives of the attacks, and X-axis shows different attack targets in

Table 5.2: Impact analysis on LMP manipulations

P.Levels / Targets	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Bids of G3	0	0	3.0	5.4	7.7	10.2	12.6	14.9	17.3	19.7
Bids of G4	2.6	5.1	7.7	10.3	12.8	15.4	18.0	20.6	23.1	25.7
Capacity of G3	0	0	0	0	51.9	51.9	51.9	51.9	51.9	51.9
Load at bus 2	0	0	0	0	0	0	0	0	0	0

market operation. Triangles on a specific row represent optimal attack targets for a specific attack objective. For example, for the attack that is to maximize the LMP at bus 1, the optimal attack targets are the load at bus 12 and the line rating at line 15. In other words, an attack on these two parameters will most effectively alter the LMP at bus 1 than the attacks on any different combination of two parameters.

Therefore, by enumerating the number of triangles on each column, the probability of being attacked can be estimated for each parameter from the perspective of being a highly probably attack target. In other words, the column that has the most triangles indicates the parameter that has a high probability of being attacked. In this study, the line rating of line 15 is the most vulnerable parameter which will be the most frequent attack target. Therefore, if this target is protected, most attacks become less effective. Although the attackers' objective is usually unknown in reality, protection of highly probable targets reduces the overall attack interest in the market operation. The upper subplot and lower subplot in Figure 5.7 mean different attack degrees (2 and 3), namely, how many parameters that the attacker is able to modify. With the attack degree increases from 2 to 3, the possibility of attacking the line rating of line 15 increases from 48.6% to 71.6%. Therefore, if the line rating of line 15 is immune from attacks, the interests of most attacks in this market are much reduced.

Identifying devasting attack targets

This study aims to demonstrate Algorithm 2. The CVA model on interested attack objectives is solved iteratively for a gradual increase of the penetration levels of different attack targets. The deviations between the objective value under normal operation and under attack are recorded. The computational time of Algorithm 2 in this study is 135.25 s. We select the most popular two attack objectives in the literature as examples: (1) diminishing the social welfare and (2) manipulating LMPs (bus 10). The impact analysis of 4 different attack targets on those two objectives is shown in Table 5.2 and Table 5.3. Simulations on other attack objectives and targets can be performed similarly. For LMP manipulation, an attack on unit 4's bid is more effective when the penetration level is low, and an attack on unit 3's capacity becomes more effective when the penetration level is higher than 40%. For diminishing social-welfare, attacking load at bus 2 is more effective when the penetration level is lower than 30% or higher than 90%, and attacking unit 3's bid is more effective for other

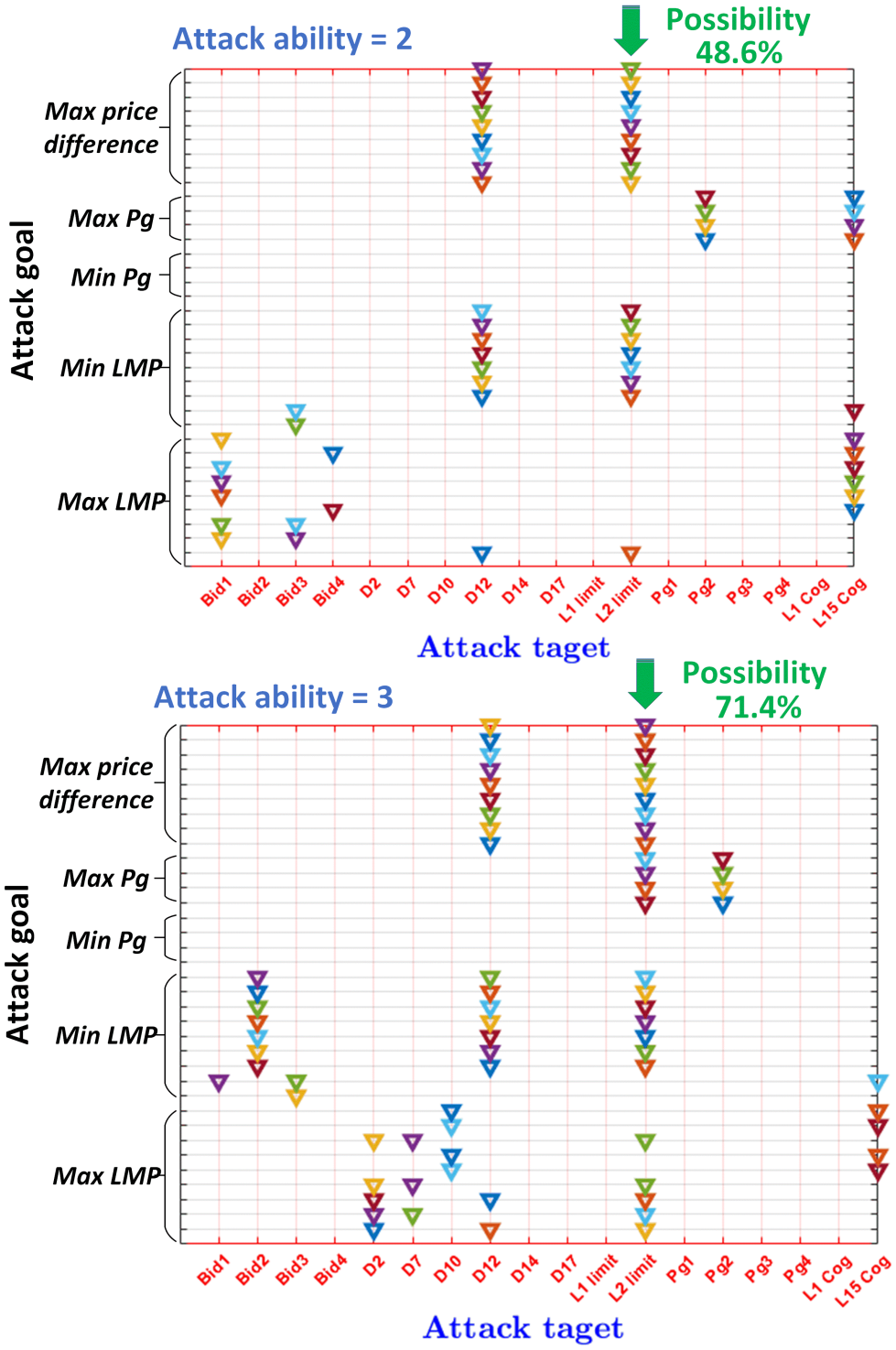


Figure 5.7: Identifying the most likely attack target

Table 5.3: Impact analysis on diminishing social-welfare

P.Levels / Targets	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Bids of G3	0	0	207.1	207.1	207.1	207.1	207.1	207.1	207.1	207.1
Bids of G4	0	0	0	0	0	22.0	22.0	22.0	22.0	22.0
Capacity of G3	1.2	3.6	5.5	7.3	9.1	10.9	12.7	14.5	16.4	18.1
Load at bus 2	30.0	60.0	90.0	120.0	150.0	180.0	210.0	240.0	270.0	300.0

penetration levels. Further, a step-change phenomenon is observed for both attack objectives. The social welfare loss exhibits a step-change pattern with the bid modification attack and continuously changes with the remaining attacks, while LMP continuously changes with the bid modification attack and exhibits a step-change pattern with the remaining attacks. The reason is that the bid modification attack does not impact social welfare unless it changes the dispatch results since it does not change the generation cost in practice, but the bids of marginal units directly impact the LMP. If the most sensitive attack target is identified and protected, the attack interests for a specific attack are significantly reduced.

Evaluating risk load levels

This study aims to demonstrate Algorithm 3. The CVA model for all attack objectives is solved iteratively under different load levels. The deviations between the objective value under normal operation and under attack are recorded. The computational time of Algorithm 3 in this study is 965.36 s. Figure 5.8 shows the risk evaluation of different load levels by a heat map. Different attack objectives have their own heat map (i.e., risk zone). Here, all risk zones are summed and scaled to be between 0 and 1, where 0 means not risky and 1 means the riskiest. Thus, the more overlap of the risk zones, the brighter the square is. That is, a brighter area means more impact to the market operation.

As shown in Figure 5.8, at first, the heavier the load is, the more an attacker can do. However, when the load becomes higher, the impact decreases because the margin for manipulation by the attacker is decreased. In other words, when more generators are at maximum, there is less room for an attacker to manipulate the parameters without being detected.

Investigating the mitigation ability of different degrees of defense

This study aims to demonstrate Algorithm 4. The understanding of how defenses improve the deviation from the optimal dispatch provides a guideline for a market operator to develop defense strategies. The CVA model is solved iteratively with a gradual increase of the defense degree. The computation time of Algorithm 4 in this study is 65.39 s. As shown in Table 5.4, the value of deviation from a normal value gradually decreases to zero with the increasing defense degree.

When more highly effective attack routes are blocked (i.e., at higher defense degrees), the attacker has to switch to less effective attack routes, and thus, the impact of cyberattacks is much more alleviated. Although the attack still impacts the market operations unless all of

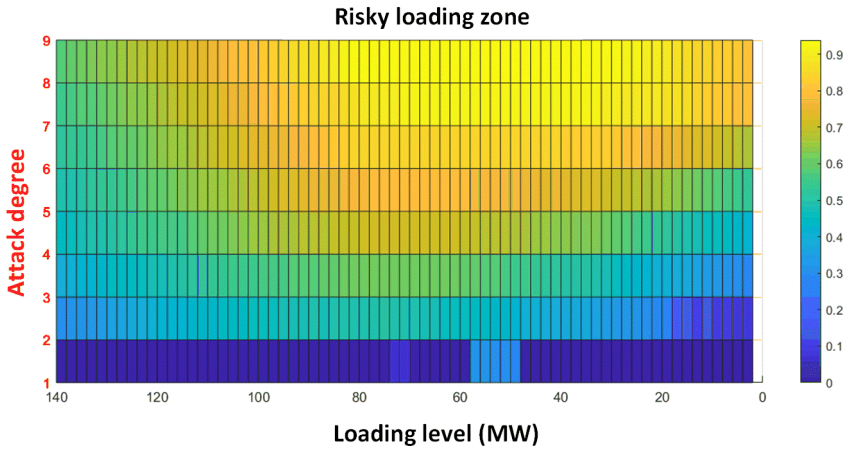


Figure 5.8: Vulnerable market operating zone

Table 5.4: Impact analysis on defense degree

Objective \ Degree	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Social-welfare loss	109.2	105.1	101.0	86.2	72.1	55.3	35.7	24.6	0
LMP (bus 10)	215.9	215.9	215.9	215.9	215.9	215.9	132.0	30.3	0

the compromised parameters are corrected, the attacker could lose interest when the degree of defense is higher than a certain threshold such that the attacker's gain from cyberattack is very low. The proposed analysis provides the market operator the information of critical defense degrees. As shown in the first row of Table 5.4, when 3 of the most effective attack routes can be protected, the maximum social welfare deviation dropped from 109.2% to 86.2%, which may discourage the attacks. Further, the social welfare loss due to cyberattacks decreases almost linearly with the increasing defense level. For an LMP manipulation attack, as in the second row of Table 5.4, the defense is not effective (i.e., the deviation created by the attack is 215.9%) until 5 parameters can be defended, which means the attackers can still achieve the desired outcome with the rest of the undefended measures. When the defense degree is larger than 5, the optimal value of the attack objective starts to decrease. It should be pointed out that the proposed algorithm provides useful information for a decision maker while the actual threshold to determine the number of defense degrees is a choice of the decision maker.

5.4 Conclusion

In this chapter, the missing components in the current research on power market cybersecurity were discussed. A cyber-vulnerability analysis (CVA) model was developed for market operator to perform impact analysis on market cyberattacks. Four vital components related to cyber vulnerability in the system were discussed with respective algorithms. The proposed algorithms can help the market operator identify highly probable attack targets, devastating attack targets, risky load levels, and the mitigation ability of different defense degrees. In summary, the CVA model developed in this effort provides a new method to identify various aspects that are vulnerable to cyberattacks in market operation, which provides valuable references for further development of cyber defense strategy.

Chapter 6

WISP Algorithms: Root Cause Analysis

ISOs in the United States typically involve multiple rounds of market clearing through centralized unit commitment and economic dispatch processes. Every market clearing process requires various types of input data, such as load and renewables forecasts, telemetry, etc. The data is usually acquired from a mix of internal and external sources, such as market participants. Hence, the market processes present, in theory, multiple avenues for malicious actors to inject false data to, a) achieve financial objectives through market manipulation, or b) steer the system into a stressful state, which can eventually lead to physical breakdown and outages.

Regardless of the malicious objective, false data injection can result in anomalous market outcomes, such as unexpected price spikes, congestion patterns, etc. Price spikes can result from a combination of factors, such as forecast errors, generation and transmission outages, etc., which can happen even in the absence of any malicious activity. Hence, proper diagnosis techniques are required to ensure that market outcomes, such as price spikes, are attributed to their true underlying cause(s). In this chapter, we will discuss techniques to analyze root-causes of historical price spikes to identify features of the input data sets, market processes, and IT systems that can be used by malicious actors to influence market outcomes.

6.1 Introduction

The operation of the electricity grid is becoming complex with increased renewable penetration, which often leads to differences in planned versus actual operations between day-ahead and real-time markets. Various market instruments like virtual bidding, reserve markets, flexible ramping products, and demand response provide mechanisms to achieve convergence between the markets and manage imbalances and uncertainties between day-ahead and real-time markets [120]. Even with advanced market instruments in place, different markets tend to encounter price differences, and often, sudden spikes in the real time market prices. Such spikes in energy price can be caused by one or more system operating conditions occurring in different temporal and spatial combinations. Figure 6.1 shows the overall

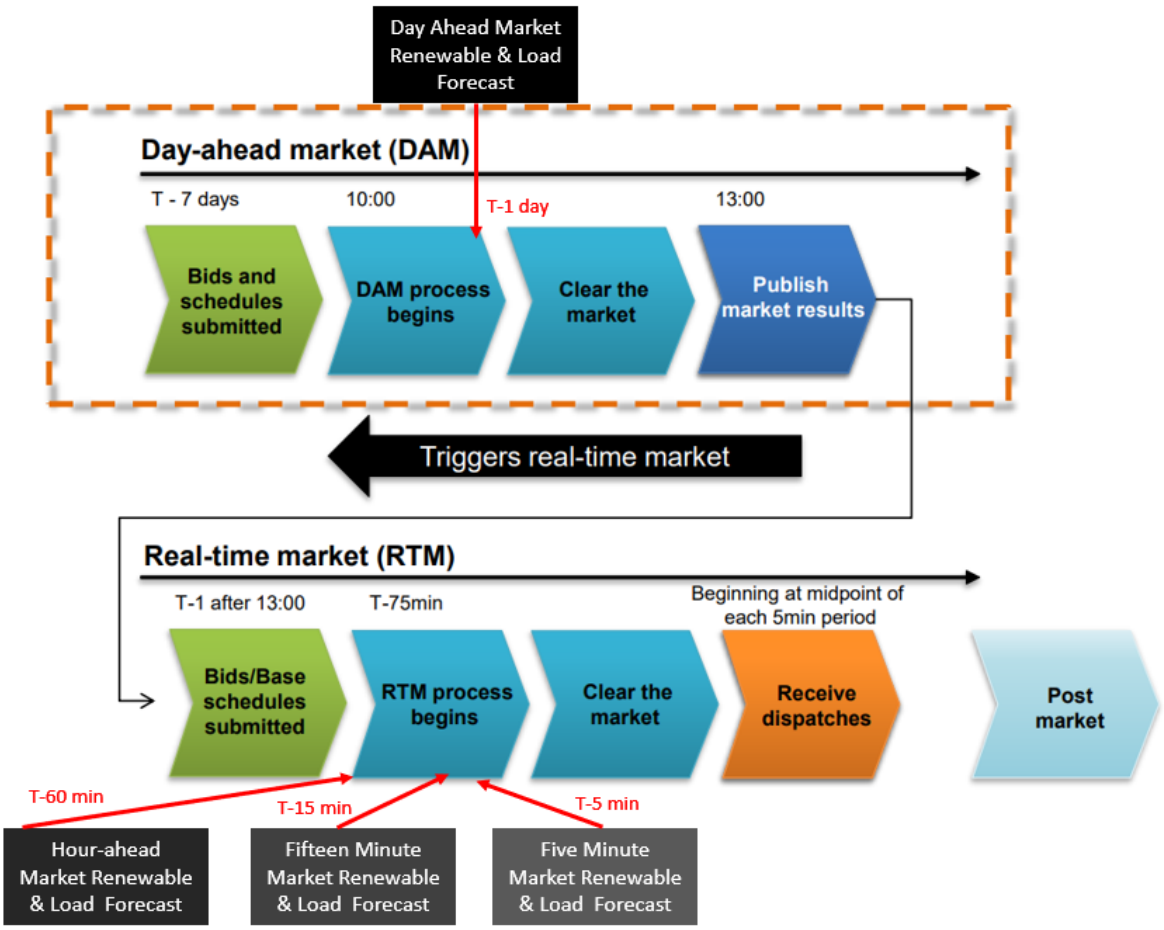


Figure 6.1: CAISO Market Architecture and Inputs at different time intervals [4].

market structure diagram for CAISO with the timelines for the different market processes. It is to be noted that the input parameters like load and renewable forecast to the market are updated at different time intervals and if compromised near real-time can result in significantly different market outcome. It is therefore necessary to understand the different state space of the system conditions which can lead to price spikes and which parameters, if compromised, can result in a price spike event.

There is a significant amount of literature including [92], [121], [122], [123] that addresses the problem of price spike forecasting in electricity markets; however, there is a dearth of research on price spike root cause analysis. While forecasting of price spikes is an important problem, it does not provide any insight on the primary drivers behind price anomalies. A root-cause analysis is not only crucial in understanding system states when price spikes might happen, but also differentiate spikes due to true underlying causes from the spikes caused

by the malicious actors to influence the market outcome. The market processes present, in theory, multiple avenues for malicious actors to inject false data to, a) achieve financial objectives through market manipulation, or b) steer the system into a stressful state, which can eventually lead to physical breakdown and outages. Regardless of the malicious objective, false data injection can also result in anomalous market outcomes, such as unexpected price spikes, congestion patterns, etc. Therefore, it is important to identify the underlying causes between price spikes, so that inappropriate market-based interventions can be identified, if not fully mitigated. Furthermore, proper diagnosis techniques are required to ensure that market outcomes, such as price spikes, are attributed to their true underlying cause(s).

The challenges in accurately forecasting price spikes, and subsequent root cause analysis, can be due to (a) rare occurrence of price spikes compared to "regular" prices; (b) complex interactions between multiple system conditions leading to price spikes; (c) limited data availability due to confidential nature of price bids and generator availability. The major contribution of this chapter is to provide a robust methodology for identifying key root causes behind price spikes and ways in which they can be manipulated by a malicious user to alter market outcome. In this chapter, publicly available energy market data from the California Independent System Operator (CAISO) is used to train machine learning models to identify probable root causes. The process uses machine learning models to discover the complex temporal and spatial interactions between 100s of system state variables, which are then used to assign a confidence level to different root causes for the price spike events. The confidence score can assist the operators in classifying price spikes due to malicious activities.

6.2 CAISO Energy Market

The root cause analysis methodology is tested using publicly available data from CAISO [124], though the methodology can be extended to any electricity market without loss of generality. CAISO manages a day-ahead market and an intra-day real-time market that economically dispatches generating resources to serve the forecast load, while managing various transmission and generation constraints. CAISO is a *nodal* market, which generates locational marginal prices (LMP) for over 4000+ price nodes throughout its footprint. In this chapter, price spikes occurring at only the four major locational aggregate price (LAP) nodes for PG&E (Pacific Gas and Electric Company), SCE (Southern California Edison), SDGE (San Diego Gas and Electric), and VEA (Valley Electric Association) are analyzed.

The root-cause analysis process requires a study of correlations between prices and the various exogenous and endogenous set of data features, such as renewable and load forecasts, etc. The raw data for this analysis, obtained from [124] for the year 2019, comprises of the following set of features for the day-ahead, hourly, fifteen-minute and five-minute markets:

1. Prices for PG&E, SCE, SDGE, VEA regions

2. Ancillary Market Prices and Cleared MW

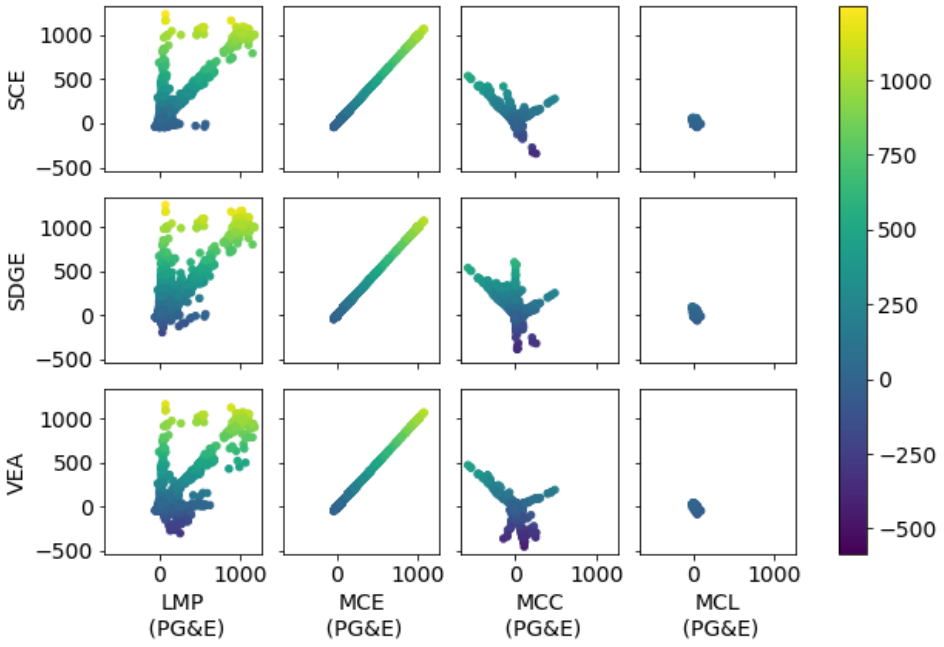


Figure 6.2: Price distribution

3. Congestion price at various flowgates and interties
4. Real-Time (5-min) Aggregate Supply/ Generation (MW)
5. Actual System Demand (MW)
6. Day Ahead and Real-Time Demand Forecast (MW)
7. Day Ahead and Real-Time Renewable Forecast (MW)
8. Area Control Error (ACE)
9. Energy Imbalance Market (EIM) Transfers (MW)

6.3 Approach to Root Cause Analysis

To illustrate the relationship between various components of an LMP, and their spatial distributions, Figure 6.2 depicts correlation between the LMP components of PG&E LAP (x-axis) and other CAISO LAPs (y-axis). LMP (for all the nodes) is a linear combination of MCE, MCC, and MCL. In this analysis, we will only focus on identifying root causes that impact the marginal cost of energy i.e. MCE.

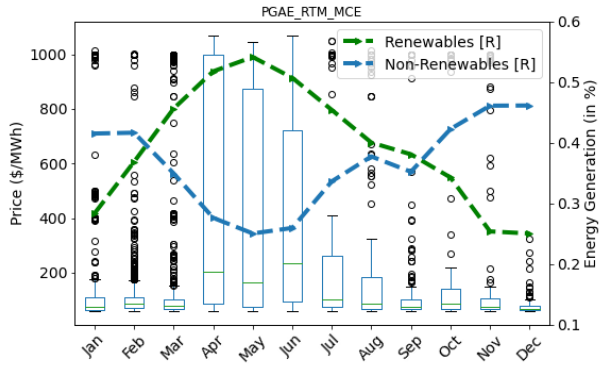


Figure 6.3: Spike distribution across months.

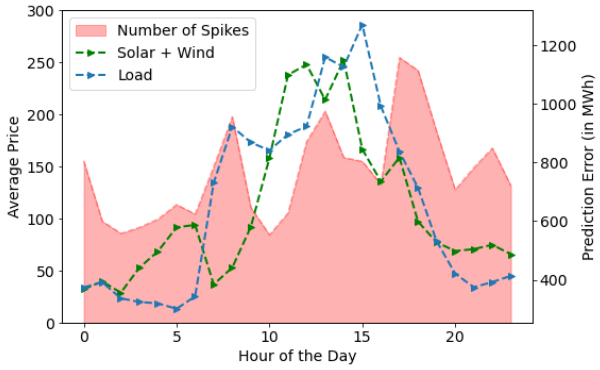


Figure 6.4: Spike distribution across HoD.

Table 6.1: State Space Representation

Feature	Mathematical Formulation	Interpretation
Standard		
mean (μ_i)	$\frac{1}{n} \times \sum_{t=0}^{n-1} X_i^{(t)}$	average value of the signal X_i during a segment
Gradient		
g_{avg}	$avg_{t\in[1,n-1]}\{X_i^{(t)} - X_i^{(t-1)}\}$	average gradient of signal X_i during a segment
g_{max}	$\max_{t\in[1,n-1]}\{X_i^{(t)} - X_i^{(t-1)}\}$	maximum gradient of signal X_i during a segment

6.3.1 Exploratory Analysis of Price Spikes

Distribution of price spikes: The process of identifying root causes begins with exploratory analysis of time series data, individually and in relation to other data features. Figure 6.3 depicts the distribution of energy costs (MCE component of LMP) across the year. The right y-axis presents the average monthly contribution of renewable sources of energy (solar, wind, hydro, and others) and non-renewable sources of energy (thermal and nuclear) towards the total energy demand.

More positive price spikes, defined here as price exceeding $150, were observed in the spring and summer seasons than the fall and winter, which coincided with the higher contribution from renewable sources as well as increased forecasted load. Similar patterns were observed from the analysis of average hourly generation for the entire year, as seen in Figure 6.4. Preliminary analysis of raw time-series data suggested a high degree of correlation between price spikes and renewable generation. The exact set of data features, which were deemed to be root causes for price spikes will be discussed later in the chapter. The process of preparing data sets for the analysis is described next.

6.3.2 Data Segmentation based on System State

The CAISO market processes begin a day ahead of actual operations, based on forecasts of renewable and system load. Subsequently, the markets clearing processes are run hour-ahead, fifteen-minutes ahead, and finally, five-minutes ahead of real-time dispatch. As the market processes near real-time operations, various inputs, especially forecast variables, become more accurate requiring changes in commitment and dispatch schedules to manage the imbalances. If the forecast values are significantly different relative to previous intervals, large-scale imbalances can manifest resulting in higher probability of price spikes in the future.

Figure 6.5 shows a specific day in the CAISO market in 2019, where renewable forecast errors were significantly high. The renewable forecast error at 08:00 led to increased imbalance causing a price spike. On the other hand, the price spike at 14:00 was due to the intermittent nature of solar energy and a drop in solar generation resulted in other resources to ramp up and balance the mismatch. Manual inspection of price spike events indicated that the cause for price spike in a given market internal had *roots* in the changes transpiring in past market intervals. Therefore, we divided the data into hour-long segments to identify key features related to a price spike event. We kept a buffer of 30 minutes between all the segments to avoid data overlap for accurate modeling. For every segment, *maximum* MCE was used as the price label and three statistics were calculated for each of the data feature, as listed in Table 6.1. The mean (μ) estimates the average value of the feature and the gradient stats compute the change in feature during that time interval. We used two gradient statistics, g_{avg} and g_{max} that report average and maximum change in the feature value. This resulted in data set consisting of over 90+ feature vectors. Hence, machine learning tools were used to identify features of greatest significance, the process for which is described next.

6.3.3 Feature Identification and Extraction

Three machine learning tools were used to find correlation between prices and the feature set consisting of both, raw data and its derivatives.

Self-Organizing Maps (SOM)

Self-Organizing Maps (SOM) [125], a computational method for visualization and analysis of high-dimensional data, was used to devise hypotheses based on visual observations of correlation patterns between the feature set, coupled with domain knowledge of electricity markets. Figure 6.6 provides an example of data visualization using SOM analysis. The SOM analysis was performed on the raw data, as well as the derivative set of features that incorporate rate of change and trend elements, using the SOM Toolbox [126]. It can be observed that price spike intervals align with periods of high thermal and wind generation

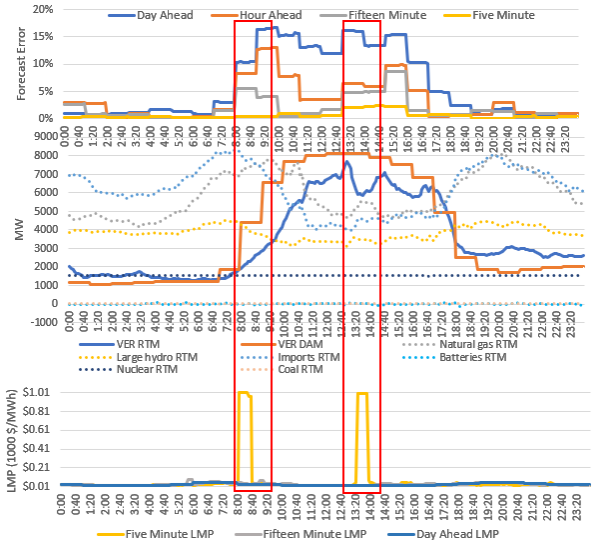


Figure 6.5: Forecast error leading to spike.

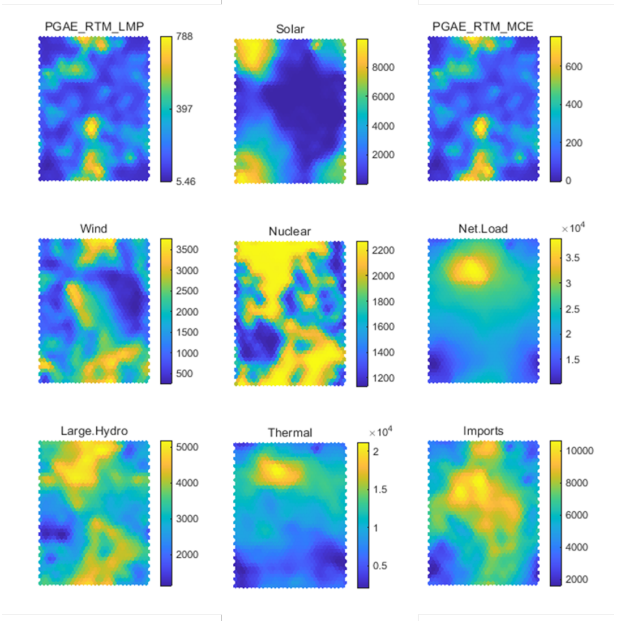
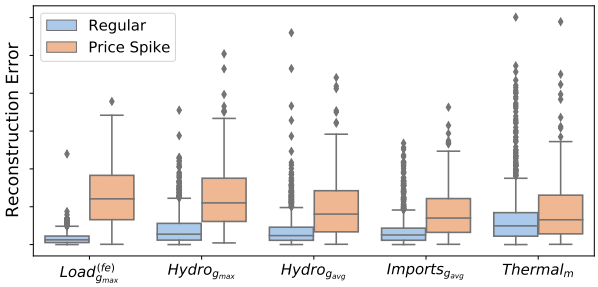
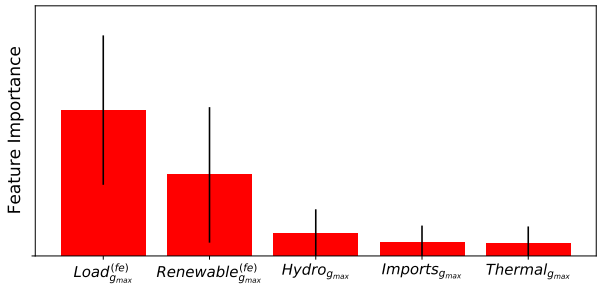


Figure 6.6: Feature visualization using SOM.



(a) Feature-wise reconstruction error from the Autoencoder.



(b) Feature importance from Random Forest classifier.

Figure 6.7: Feature Extraction using Autoencoders and Random Forest.

levels, whereas few spikes occur when solar generation is high. From SOM analysis, the following feature sets were identified to be strongly correlated to price spike intervals:

- Renewable forecast errors
- Demand forecast errors
- Regulation-up price spikes (proxy for generation outage)
- High Load/ Net Load state

- High Load change/ Net Load change
- Low Solar generation state
- High Wind production
- High Thermal production

It should be noted that SOM analysis does not provide a quantitative measure of correlation. Testing out the hypotheses devised from visual inspection of SOM results requires additional analysis, such as using *autoencoders* and *random forest classifiers*, presented next.

Autoencoders

Autoencoder [127] is an unsupervised deep learning technique used to learn a low-dimensional latent representation of the high-dimensional data. The latent representation is then used to recreate the input feature set. The reconstruction error between the input and the output feature set is a key metric to identify features correlated with price spikes. The process requires training the autoencoder using the non-spike data segments and then passing the spike data segments through the autoencoder. The resulting set of the reconstruction errors are likely to be high for those features which are highly correlated with the price spikes.

Figure 6.7a compares the reconstruction error for top five features (sorted by reconstruction error in price-spike segments) when the trained autoencoder is used to estimate input feature space for price spike segments v/s non-spike segments. It is evident from the plot that the reconstruction error is significantly higher for gradient based features like g_{max} and g_{avg}, whenever there exists a price spike. The reconstruction error for mean based features are similar for both non-spike and spike events.

Overall, the analysis highlights a greater significance of gradient-based features, such as *rate of change* of forecast error over mean-based features, such as *average* forecast error over a time window for root-cause analysis.

Random Forest

Next, a random forest classifier was trained (in scikit-learn [82]) to further quantify the importance of features. The classifier takes feature values as input and classifies each segment as a regular vs. a price-spike event. Random forest [128] is an ensemble machine learning technique that uses multiple decision trees to predict a class, and the class with maximum votes is predicted as the outcome. In every decision, the data split happens based on feature values, and the quality of every split is evaluated based on either *gini impurity*

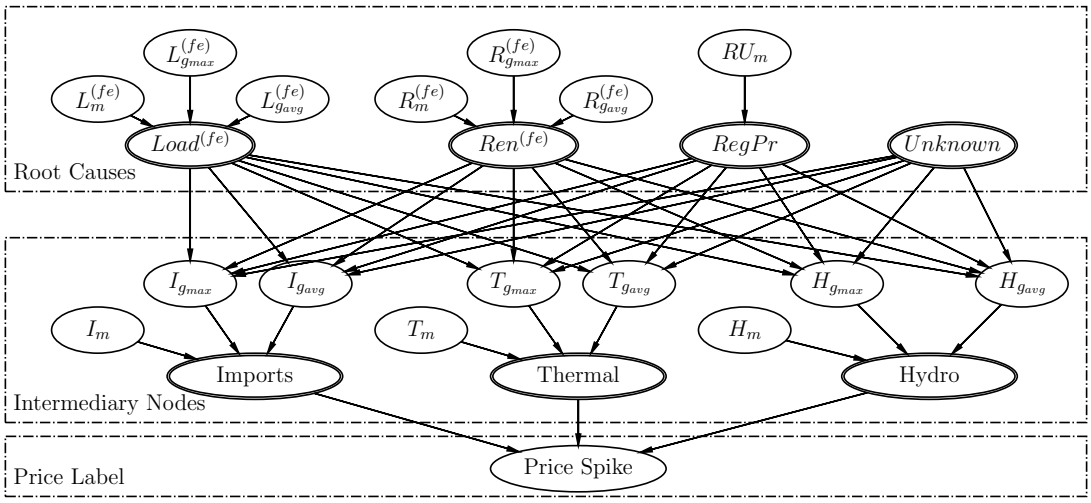


Figure 6.8: Graphical model for the root cause analysis of the price spike events.

or *entropy*. Figure 6.7b depicts the distribution of gini impurity (or also known as feature importance) of top five features, sorted by their importance (left to right shows most to least important feature) for the trained classifier. The results are in agreement with the insights from autoencoders.

These observations led to three major conclusions: (1) The price spikes are highly correlated with the forecast errors, both in load and renewables. (2) The gradient-based features are highly correlated with the price spikes. (3) The mean-based features are critical to identify if the change in any feature is significant enough to cause a price spike. Based these insights, the following 16 features were chosen for Bayesian Inference modeling.

- Load Forecast Error: $L^{(fe)}_{m}$, $L^{(fe)}_{g_{max}}$, $L^{(fe)}_{g_{avg}}$
- Renewable Forecast Error: $R^{(fe)}_{m}$, $R^{(fe)}_{g_{max}}$, $R^{(fe)}_{g_{avg}}$
- Regulation-Up Prices: RU_m
- Imports: I_m, $I_{g_{max}}$, $I_{g_{avg}}$
- Thermal: T_m, $T_{g_{max}}$, $T_{g_{avg}}$
- Hydro: H_m, $H_{g_{max}}$, $H_{g_{avg}}$

6.4 Bayesian Modeling

Next, the key takeaways from the exploratory analysis were used to build the Bayesian Graphical Structure (as shown in Figure 6.8): a directed acyclic graph where nodes indicate the features and the edges between them indicate dependency of one feature on the other. The node from where the arrow begins is the *parent node* and the node where it ends is the *child node*. For each node, we compute conditional probability distribution (CPD) table from the data [129].

6.4.1 Structure Representation

In the current implementation (Figure 6.8), the graphical structure is divided into three sections. Root causes include load forecasting error, renewable forecasting error, and change in regulation up prices. The segments for which we couldn't assign a root-cause, we put them under *Unknown*. Any significant change in these features might force the regulators to bring in more resources in the form of thermal, imports, and hydro: the intermediary causes. These nodes can make direct impact on the price label - spike or non-spike. To decide if a feature caused the price spike, the model evaluates the feature values. For example, to decide if the load forecasting error ($Load^{(fe)}$) was one of the root-causes for the price spike, the model will examine its mean value and change in its value just before the price surge.

6.5 Evaluation

For each query, the trained model takes the value of child nodes as an input and estimates the probability of any node being the cause for price spike. The framework then translates these estimations into a human-readable explanation. To illustrate, in one scenario, the model estimated *Imports=True*, *Thermal=True*, and *Hydro=False* knowing that *Price Spike=True*. The framework used this information to next evaluate corresponding feature values ($I_m, ..., T_{g_{max}}, ..., H_{g_{avg}}$) to understand *"what went wrong?"*. The model found that $T_{g_{max}}, T_{g_{avg}}, I_{g_{max}}, I_{g_{avg}}$ ramped up during that period. The model then used these feature values to estimate labels for the parent nodes: $Load^{(fe)}$*=True*, $Ren^{(fe)}$*=True*, and *RegPr=False*. Eventually, the framework generated the following explanation:

> *With 92% confidence, the price went up because thermal and imports ramped up significantly. With 95% confidence, thermal and imports ramped up because mean load forecast error was high and renewable forecast error jumped up significantly.*

In total, 187 price spike events and 6,237 regular price segments were recorded in the data. In manual labeling, load forecasting error was the root-cause in 87% cases, renewable forecasting error was in 71% cases, regulation prices up was in 11% cases. Likewise, in intermediary-causes, thermal was the intermediary-cause in 75% cases, imports in 89% cases,

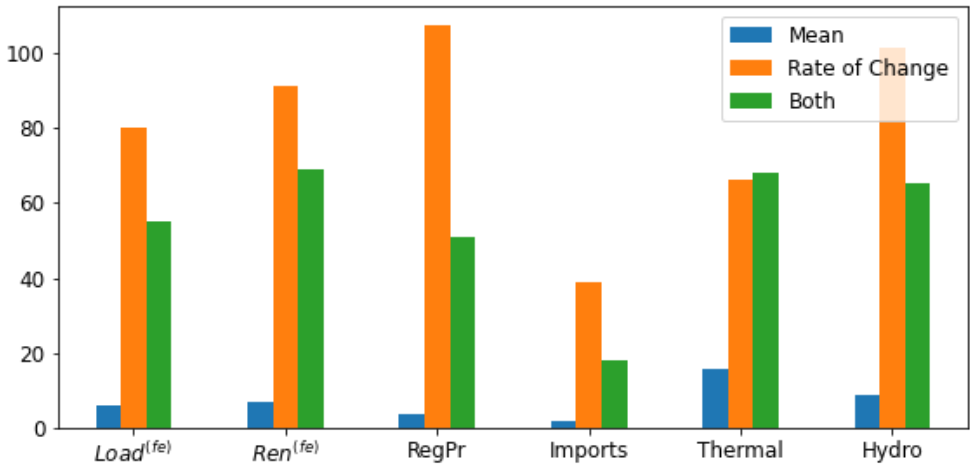


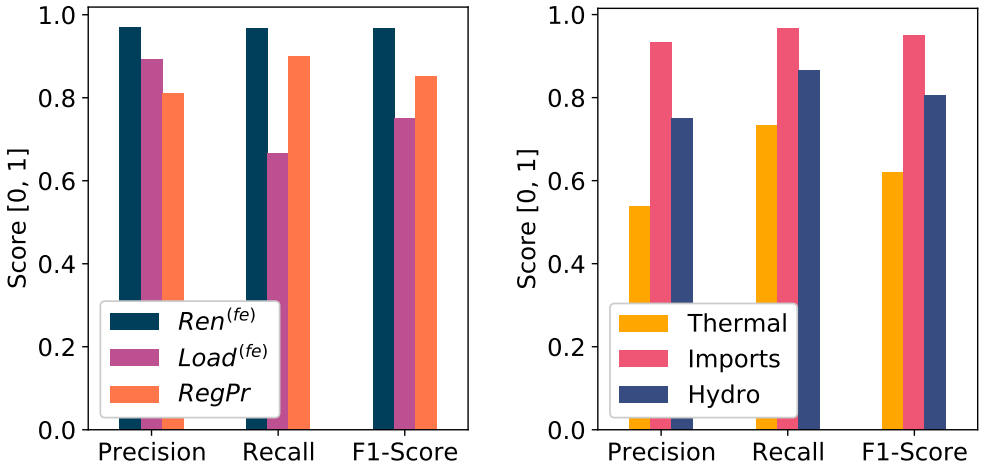
Figure 6.9: Spike distribution for different reasons.

and hydro in 87% cases. Figure 6.9 further shows distribution of the price spike instances across the key reason for every root-cause and intermediary cause. It is evident from the plot that spikes either happen due to sudden change in the signal only, or sudden change in the signal along with a high mean value. In only a few cases, high mean value of the signal alone is the root-cause for the spike.

Since change in the signal is a leading cause, one can easily infect a few data points to generate unintended market outcomes. For instance, by deliberately modifying the renewable and load forecast in the day-ahead market, the system can be compromised to generate price spikes in the real-time market. Based on the above-mentioned analysis, following is a list of possible cyber-attacks in the energy market.

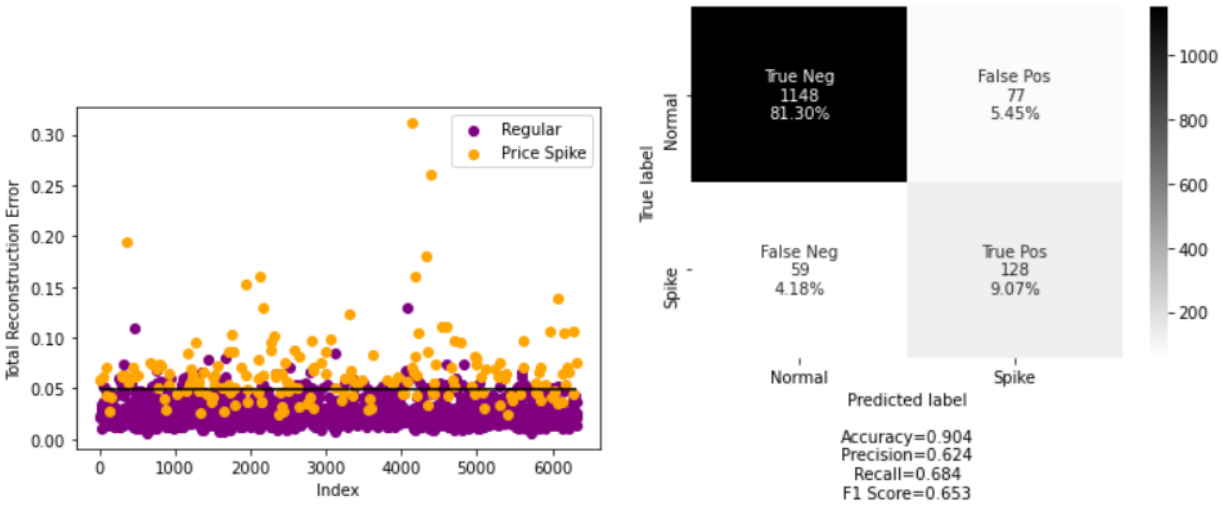
- Malicious Forecast: In such scenarios, one can intentionally keep the load and renewable forecast value high/low for a small period to introduce price spikes in the market.
- Data Modification Attack: In such attacks, a malicious user can change the real-time data streams to introduce a sudden change in the signal, and thus the price spikes.

The Bayesian model, as of now, cannot identify both above-mentioned attacks because it assumes that the incoming data is correct and uncorrupted. The model was trained on 100% regular segments + 70% price spike segments, and tested on remaining 30% price spike segments. Figure 6.10 shows the model accuracy based on the scales of precision, recall, and F1-score in identifying the root-causes and the intermediary causes. The results indicate that, on an average, the proposed model can identify root causes and intermediary causes with an accuracy of 86% and 80%, respectively.



(a) Root Cause Nodes (b) Intermediary Nodes

Figure 6.10: Accuracy numbers for the test data.



(a) Anomaly score for all instances. (b) Confusion matrix.

Figure 6.11: Classification based on autoencoders.

6.5.1 Suspicious Instances

Figure 6.11a shows the reconstruction error for a batch of regular and spike instances as tested by the Autoencoder model. For non-spike (or regular) events, the error is usually low, and high for the spike events. The black line in the plot should ideally separate regular from price spike events. While some events are far apart, several regular and price spike

Table 6.2: Similar state space representation for a regular and a price spike event. Here, Renewable implies Solar and Wind.

Feature	Regular Price	Price Spike
Price	**35.177190**	**952.562400**
Imports Median	8623.510663	8711.206627
Imports Standard Deviation	103.771116	325.678216
Imports Trend	282.402748	788.843837
Thermal Median	16446.598150	16463.263180
Thermal Standard Deviation	331.121176	282.372849
Thermal Trend	830.156730	825.002810
Renewable Median	5578.540764	3527.708979
Renewable Standard Deviation	772.450192	759.682878
Renewable Trend	-2059.053798	-2079.383630
Renewable_Forecast_Error Median	-915.200000	-1858.320000
Load Median	39785.664840	37823.860630
Load Standard Deviation	239.247281	161.397481
Load Trend	-663.090910	-454.278190
Load_Forecast Median	39427.000000	37538.000000
Load_Forecast_Error Median	-1558.225160	97.100630

instances lie around the line. These instances are the confused events (or suspicious events) that require manual verification.

As shown in Figure 6.11b, on a randomly sampled test data, the model found state space representation of the 77 instances (out of 1225 regular price events) similar to the price spike events, however, no spike was noticed in those time periods. Likewise, the model noticed 59 instances (out of 187 instances) where the state-space representation indicated regular price, but price spike occurred in the actual data. While a few of them may be attributed to the modeling error, 10 (orange points in Figure 6.11a very low reconstruction error) belonged to the *Unknown* category in the manual labeling and thus require more data and further evaluation. Table 6.2 depicts once such scenario where even though the state space representation is similar, market noticed spike in one scenario and no price spike for the other scenario.

6.5.2 *Unknown* Spikes

Once the true reason is identified by the Bayesian model, it is easy to differentiate scenarios for which it is hard to pinpoint a root-cause because everything looks normal to the model. In the current implementation, all such instances lie under the *Unknown* category. In the described dataset, 10 such spikes were noticed where it was hard to manually find the root-cause for the price spikes.

6.6 Conclusion

As the electricity grid continues to evolve with increases in renewable penetration, electricity markets will continue to see changes in price behaviors - price spikes, volatility, and negative prices. Root cause analyses for these price behaviors are usually done by ISOs and market monitors who have access to sensitive data, using proprietary model-based simulations [120]. These existing techniques are not usually automated, and hence, require significant time to investigate. Besides, with more data coming in, it would only become harder to manually differentiate price spikes due to true cause(s) from malicious attacks.

The proposed machine learning-based approach provides a fast and robust methodology to automatically identify the primary drivers for price spikes using publicly available data only. Furthermore, the approach does not require the use of proprietary model-based simulations. The raw data set used to identify root causes behind price spikes in CAISO market consisted of load and renewable forecasts and their errors, etc. Machine learning algorithms like SOM, Auto-encoders and Random Forest were used to identify the data features that had significant impact on market outcomes, resulting in price spikes. These analyses helped conclude that price spikes are highly correlated with renewable and load forecast errors, and that gradient-based features, such as rate-of-change of forecast errors, tend to have greater significance in explaining price spikes than averages. From these analyses, it was also observed that the price spikes often result from complex interactions between various data features, each with their own signatures that evolve over time.

Hence, the inferences from the exploratory analysis was used to hypothesize the structure of Bayesian Graphical models, which were then trained to automatically identify root and intermediary causes for price spike events. Our evaluation of 2019 year-long CAISO data indicates that the proposed model can correctly identify root-causes in 86% cases (renewable forecast errors and load forecast errors), and intermediary causes in 80% scenarios (thermal/imports/hydro generation changes). Since gradient based features play a key role in causing price spikes, further analysis indicates that attacks like malicious forecast and data modification attack can easily be carried out by deliberately modifying renewable and load forecast in the day-ahead market.

Chapter 7

WISP Software Development

7.1 Software Architecture Design

WISP is an energy market monitoring tool that uses public energy market data to detect potential cyber-attacks on the system. Real time and simulated market data is analyzed by various diagnostic algorithms and the resulting detection data and events are stored in a database to be displayed in a user friendly interface. The detection algorithms have been developed as well as an advanced event simulation system. This document contains an overview of potential architectural components of WISP including data formats, mechanisms used for data storage and the user interfaces. In using this document, system use-cases and user preferences should be captured as this will impact the design of overall system architecture and the behaviors of the individual components of the system.

7.1.1 System Overview

The high-level components of the WISP detection system are illustrated in Figure 7.1. The system is envisioned to be a cloud-enabled or on-site tool that monitors real-time power grid data, looking for anomalous data signatures that could indicate a cyber-attack event. Power grid data is received and stored by the system and the data is analyzed and visualized in real-time. There is also a configuration and maintenance interface for tasks such as analyzing past events, retraining/tuning the detection algorithms, or inspecting any system logs. The "Real-time estimators" block contains the core WISP detection logic, much of which has already been prototyped in Python. As the majority of the system data is in the form of time-series, the "Visualization UI" could be implemented using a suitable tool such as Grafana. The "Data management" component stores all raw and processed data as well as configuration / tuning data for the estimators. The "System control/config" component is the interface used to setup and run the detection system as well as updating the estimator tuning / parameters.

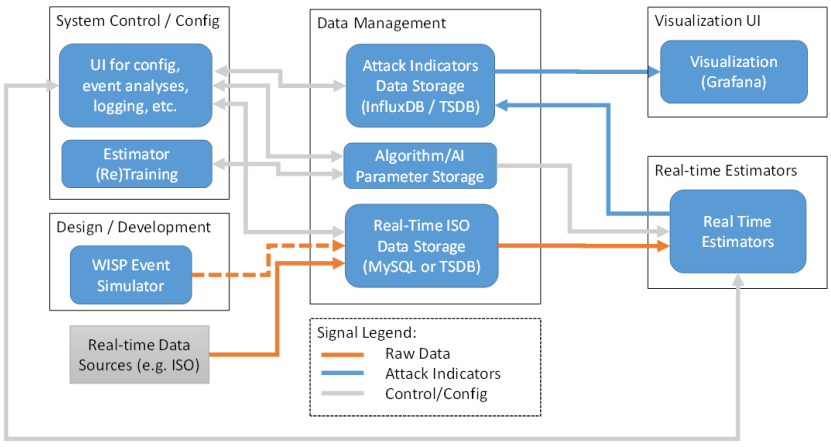


Figure 7.1: High-level WISP System Architecture.

Many frameworks are available which can be used to implement the various WISP system components. Generically, the WISP system is a service delivery framework (SDF), but specifics about the framework and platform (e.g. cloud vs. on-site) should be discussed in some detail before committing to specific solutions.

7.1.2 Component descriptions

The individual components of the WISP system are described below. Most of these components require detailed design work before any significant system implementation occurs.

Data Management:
Real-time ISO data storage. Public real-time data from providers such as ISO NE will be periodically downloaded and stored in a local database. A current prototype using MySQL database has been created but a time-series database such as InfluxDB might be more appropriate. The database schema design should consider the storage of data from multiple independent data sources, including the WISP simulator, and should then be documented. Attack indicator data storage. The WISP detection algorithms output scalar anomaly indicators as each new data set is processed. These indicators should be stored in a time-series database such as InfluxDB to facilitate visualization. The database schema design should include the ability to discriminate between different indicators to allow for the addition of new indicator types to the system. Algorithm parameter storage. The WISP estimation / detection algorithms implement various techniques for anomaly detection. Each algorithm has various parameters (tunings, trainings) that must be placed in permanent storage for retrieval during system startup. For example, training runs of AI-enabled algorithms return complex data that should be saved for later use in real-time. In most cases, the parameters will be Python data/arrays that can be stored in HDF5 files. The format, location, and

naming strategy of the various parameter files must be designed and should consider future expansion of WISP with new/different detection algorithms.

WISP Simulator:
The WISP event simulator is a MATLAB-based system for synthesizing artificial cyber-attacks. The data is then used to validate the various WISP detection algorithms. In the prototype system, this simulation data is currently exchanged via MATLAB "mat" files. The data should eventually be stored in the real-time MySQL/InfluxDB database along side the real raw data.

Real-time estimators (detectors):
The WISP detection algorithms are the central feature of the system. Various algorithmic components (e.g. feature extraction, anomaly detection, localization) are implemented to detect potential cyber-attacks. The algorithms include AI-enabled inference logic that has been trained using real and simulated data. At startup, the algorithms read their parameters from the data management system. Real-time or historical data is then retrieved from the database and processed by the detection algorithms. The various indicator outputs are then published to the indicator database for subsequent visualization.

Visualization UI:
The visualization UI is the main dashboard for viewing the real-time indicator data. Real-time or historical indicator data is retrieved from the database and displayed in appropriate time-series graphs. The UI may also be used to view raw (input) data from the database. Configurable threshold alarms and notifications are also desired. Grafana is an obvious candidate for this UI component.

System control and configuration:
The system configuration and control interface should control all other aspects of the WISP system. This includes system startup, enabling and updating (tuning) of the different estimators, analyzing current or past detection events, and other system maintenance tasks. Tools to maintain the WISP databases will be needed as there may be old records that need to be removed, or old/invalid indicator data that is no longer needed. Also, other external data may need to be imported (e.g. prior/historical data for new real-time data sources). Tools to retrain (tune) existing estimators is needed as indicator accuracy may change over time due to changing data trends. Retraining existing estimators using new data will be needed. Adding new estimators may also be needed. Appropriate update processes must be considered in the design of the control UI.

7.2 Software Development

In this section, we introduce the implementation details of the WISP software. The software stack is comprised of a data plane, a backend, and a frontend. Figure 7.2 summarizes these three components.

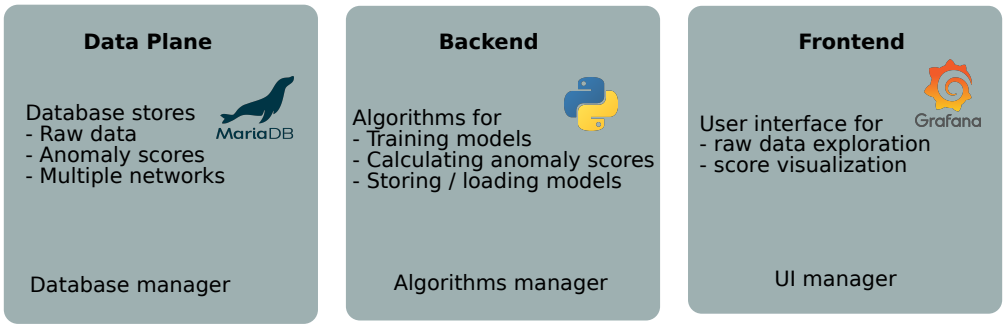


Figure 7.2: The software stack includes a data plane, a backend, and a frontend.

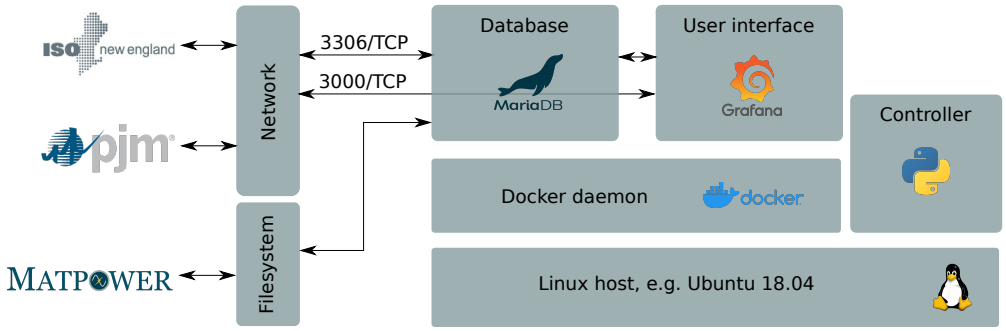


Figure 7.3: WISP's components are deployed on Linux–based host.

- The *data* plane consists of a database and a management class. The database stores raw data and anomaly scores while the management class deploys the database, loads it with raw data, and provides data warehousing services to the backend.
- The *backend* includes anomaly detection algorithms. The backend handles tasks that include training of new models, storing and loading pretrained models, and computing anomaly scores.
- The *frontend* consists of a management class and a user interface. The user interface is browser–based and provides a way for the end–user to inspect raw data, anomaly scores, and focus on selected nodes of the power grid. The management class configures and deploys the user interface.

For deployment, Docker containers are used. Specifically, each component is deployed in a separate container, a lightweight virtual machine that provides isolation from existing software on the host and robustness. Figure 7.3 shows how WISP's modules are deployed and communicate with each other.

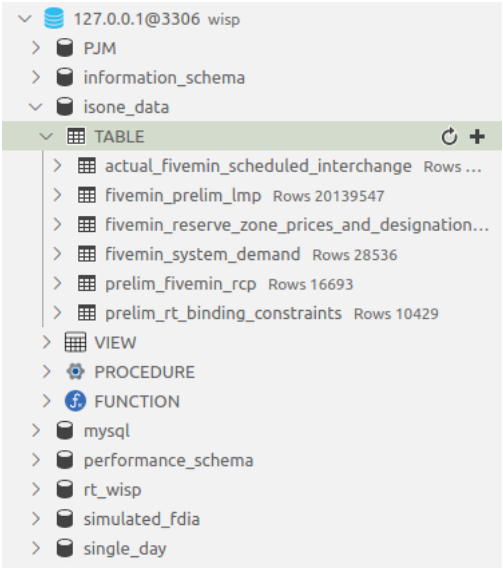


Figure 7.4: An example database holding raw data downloaded from ISO-NE.

7.2.1 Data Plane

A database and its management class comprise the data plane. For a database, MariaDB, a drop–in replacement of mySQL is used. As depicted in Figure 7.4, data from different source, including the MATPOWER simulator, ISO-NE, and PJM, are stored in different databases in their native format. Additionally, a Python class performs data warehousing by implementing several functionalities.

- *Deployment:* An empty MariaDB Docker image is downloaded from Docker Hub and deployed by the local Docker daemon. Configuration such as credentials and networking is provided during the deployment via environment variables.
- *Data downloading:* Tables are initialized and data are loaded in the database. MATPOWER raw data are loaded from an external file, whereas ISO-NE and PJM data are downloaded using the data providers' APIs. To accelerate the time from deployment until meaningful anomaly scores are produced, recent historic data are initially downloaded for training. Then, data are downloaded in real–time as soon as they become available, typically every 5 minutes.
- *Data serving:* Data for a given time window are served to the backend in a tabular format. Data from each source are formatted differently, and, thus, the data plane manager converts the data in a common tabular format on–the–fly.
- *Utilities:* The class provides supporting such as managing the connections to the

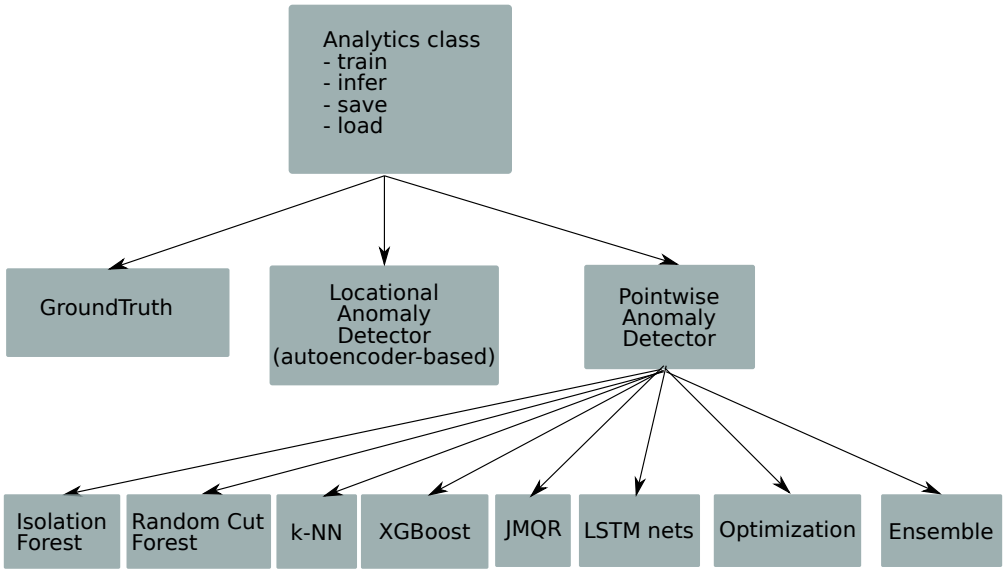


Figure 7.5: The inheritance diagram of the available analytics algorithms

database to ensure good performance and executing arbitrary SQL scripts for development and debugging.

7.2.2 Backend

The backend is the WISP's main module that processes raw LMP and load data into meaningful anomaly scores. The backend provides several deep learning and traditional machine learning anomaly detection algorithms that can be trained on historic data. The historic data is assumed to be normal and free from attack, yet, a small contamination by anomalous data is acceptable. Each anomaly detection algorithm implements at least the functionality of saving and loading an already trained model, train a new model given parameters such as time window of training data, learning rates, etc, and computing a single or a vector of anomaly scores for a given time. Each algorithm may implement additional functionality such as incremental training and focusing on a selected subset of nodes. Figure 7.5 lists all the anomaly detection methods and their inheritances.

7.2.3 Frontend

The frontend provides the interface the user interacts with. It allows the user to focus on a specific node of the network and any time window, inspect the LMPs, system load, and their components, and the calculated anomaly scores. The frontend module consists of Grafana and a management class. The management class deploys a Dockerized https://hub.docker.com/r/grafana/grafana, configures its credentials and networking

Figure 7.6: A dashboard provisioned to Grafana showing data over a whole day.

via environment variables, and provisions with a datasource and multiple dashboards. Provisioning a datasource points Grafana to the MariaDB deployed earlier and provisioning the dashboards defines the graphs to be shown to the user.

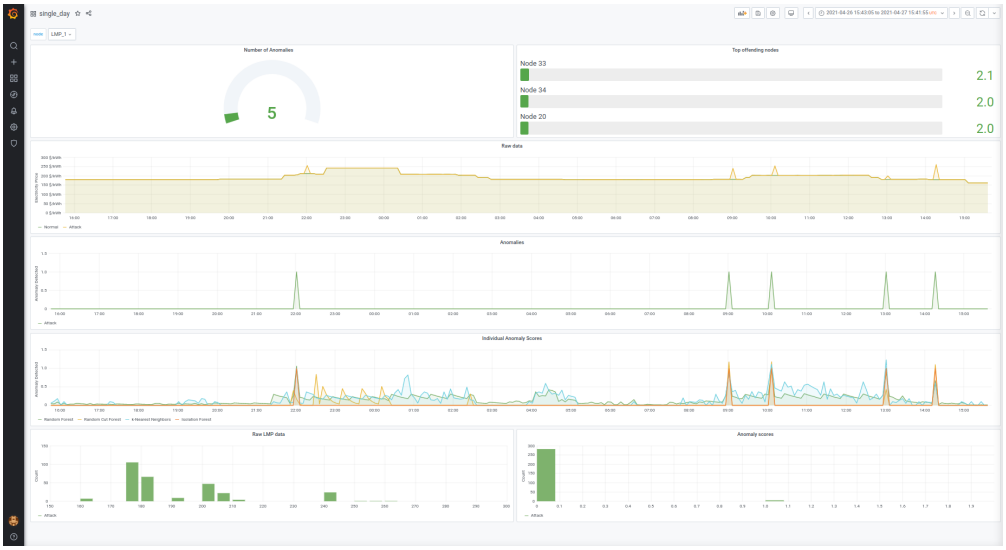


Figure 7.7: A dashboard provisioned to Grafana showing the last 8 hours.

Chapter 8

Commercialization Plan

Cybersecurity has been a major concern for electricity delivery systems, especially when transitioning to the smart grid schema. The tightly coupled cyber and physical infrastructures challenge the traditional way of defending power systems against cyber criminals. Though protected by the industry standards, the grid is still falling behind the cybersecurity technology frontiers. Given these facts, we investigated the potentials of WISP, an innovative cyber monitoring tool, in advancing the cybersecurity practices of the power industry. We surveyed the market opportunities, competitive landscape, possible commercialization paths, and capabilities of our product and our company in carrying out influential marketing activities. The findings are summarized in this chapter, including descriptions of industry standards, dominant security products, and three main deployment strategies and their corresponding target customers and cost analysis. This chapter will be updated during the course of this project as additional information and insights are gained.

8.1 Market Opportunity

8.1.1 Increasing Growth of Cyber Attacks Targeting on Electric Grids

Recent reports and surveys show that the energy sector is constantly under new, targeted, advanced and dangerous cyber-attacks that have the potential to result in the loss of human life. Examples include advanced cyber-intrusions such as the BlackEnergy, Havex, and Sandworm [130–132] malware variants that targeted critical electric power infrastructure cyber assets, including Supervisory Control and Data Acquisition (SCADA) systems. The threats against critical infrastructure from criminal groups, hackers, disgruntled employees, nation states and terrorists, whether targeted or opportunistic, are evolving and growing (see incidents reported by the Industrial Control Systems Cyber Emergency Response Team (ICS CERT) [133]). DRAGOS's report [134] identified a recent increase in cyber activities target-

ing North American electric entities, including PARISITE's VPN attacks, MAGNALLIUM's password spraying campaigns and XENOTIME's supply chain compromises. Additionally, a detailed report on the spear phishing campaign was published [135] to raise the attention of asset owners and operators. In response to these attempts, the U.S. military launched cyber attacks into the Russian power grid reported by the New York Times in 2019 [136]. Securing the electric grid was mentioned as the top priority for the U.S. federal government [137].

The threat to the electric sector is further exacerbated by the need to modernize the grid. As current power systems advance from a macro utility-centric model to a distributed structure, driven by the energy revolution, several new schemes such as smart metering, real-time pricing, managing demand side flexibility and distributed renewable energy resources, shall come to fruition. Such technologies will no doubt improve the operation of the grid and the efficiencies of the associated markets. On the other hand, it will also increase system exposure, providing newer entry points for hackers to disrupt grid operation. Based on the U.S. Government Accountability Office (GAO) report [138], the electric grid is becoming more vulnerable, especially considering the increased number of attacks on the Industrial Control System (ICS) devices widely adopted in power grids. Figure 8.1 shows the increasing number of ICS attacks.

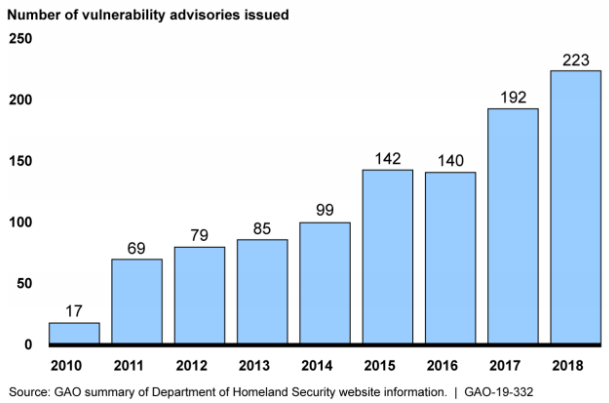


Figure 8.1: Increased Number of Vulnerability Advisories for ICS Devices

8.1.2 Solution Gaps

The cyber security threat to the energy sector is not new as the U.S. Department of Energy (DOE) has led strategic road mapping activities to address cyber security threats and improve cyber resilience since 2004. DOE's Electricity Advisory Committee recently announced the establishment of the Grid Resilience for National Security (GRNS) Sub-committee focusing on identifying and mitigating the cyber threats in energy sectors [139]. The energy sector has also made significant strides in protecting the critical cyber assets at power generation facilities through the development and enforcement of standards such

as Critical Infrastructure Protection (CIP) by the North American Electric Reliability Corporation (NERC). CIP has enforced 11 cybersecurity regulations (CIP-002-5.1a, CIP-003-7, CIP-004-6, CIP-005-5, CIP-006-6, CIP-007-6, CIP-008-5, CIP-009-6, CIP-010-2, CIP-011-2, CIP-014-2). Meanwhile, CIP keeps updating its standards to cover the areas exposed to or targeted by cyber criminals. For example, a few new standards came into effect in 2020 and 2021, listed below:

To enforce cybersecurity best practices, the Federal Energy Regulatory Commission (FERC) has performed multiple regulatory activities, such as approving mandatory cybersecurity standards, enforcing regulatory requirements and auditing NERC and bulk power entities for compliance with standards. Major transmission system operators (TSO) have released their cybersecurity strategy to the public, including compliance with the CIP standards, establishment of a security working group, and investment in cyber security hardware, software and personnel resources. Figure 8.2 shows the capital and operational cost of ISO New England to keep compliance with cybersecurity standards [140].

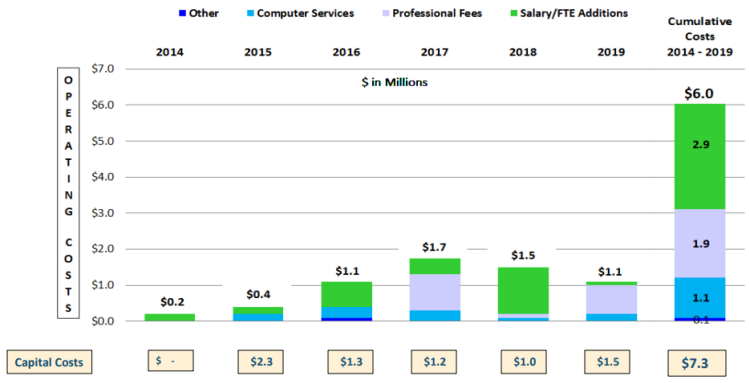


Figure 8.2: ISO-New England Capital and Operational Cost for Cybersecurity and CIP Compliance

However, based on GAO's assessment [138], the FERC-approved standards do not fully address leading federal guidance for improving critical infrastructure cybersecurity. Mapping the CIP standards to the NIST Cybersecurity Framework shows CIP standards partially address or do not address 15 out of 23 categories. NERC emphasized the CIP standards have to be industry specific and auditable, thus they cannot be one-to-one aligned to the NIST framework. Therefore, beyond the obligated regulations, entities in the energy sector are encouraged to establish their own program to manage cyber risks and mitigate cyber impacts.

8.2 Product Description

RTRC, in collaboration with the University of Tennessee and the Pacific Northwest National Laboratory, developed an advanced cyber security monitoring tool based on electricity market behaviors and using only public-available data to identify potential cyber attacks and attack targets. This technology, under the sponsorship of the DOE Cybersecurity for Energy Delivery Systems (CEDS) program, is called Watching grid Infrastructure Stealthily through Proxies (WISP). The key objective of WISP is to leverage the advanced data analytics algorithms on the vast majority of electricity market data, especially the Locational Marginal Prices (LMP), to detect anomalous prices caused by potential cyber-attacks and reason over these anomalies, in order to provide additional situational awareness to operators. To achieve this goal, the team developed three WISP modules, shown in Figure 8.3.

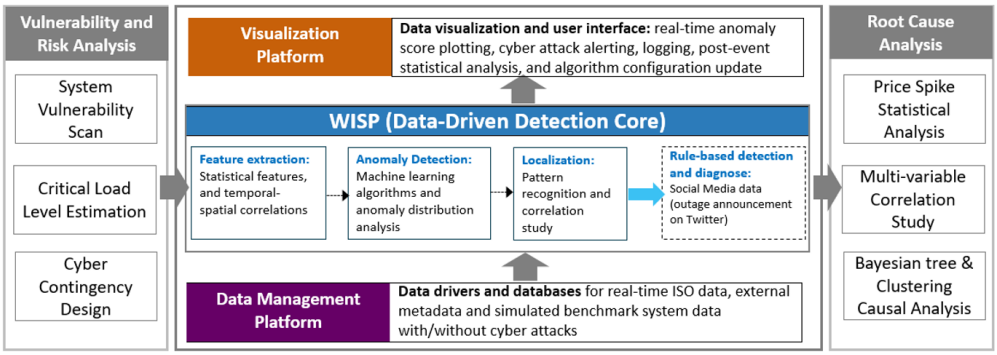


Figure 8.3: WISP framework

WISP Data-Driven Detection Core

The WISP Core provides the main functionalities related to the anomaly detection and localization tasks. It consists of three layers: the data management platform, the data-driven detection core and the visualization platform. The detection pipeline is as follows. Firstly, the customized data drivers and adapters download electricity price and system operational data from utility websites and store them into local databases. The accumulated data are passed to the detection core to train the machine learning algorithms and the trained models are then used on the real-time data stream to identify data outliers and their locations. Finally, the detection results are reported in streaming curves and statistics to the operators to raise cyber alerts and provide information for further action.

WISP Vulnerability and Risk Analysis

The WISP vulnerability and risk analysis module provide options for overall cyber vulnerability scanning of the power grid under protection. This module implements all possible attack strategies into the mathematic formulation of the electricity market management system to calculate the attack impacts under different system conditions, thus to identify the most-profitable attacks and easiest-achievable attacks. For large-scale systems, this module

helps narrow down the cyber monitoring locations and times.

WISP Root Cause Analysis

Often, the real-world market data contain uncertainties stemming from factors such as energy consumers, renewable generation, device maintenance, and extreme weather conditions. The WISP root cause analysis module provides additional input to the operators to decompose the alarms and identify major contributors to the anomalous price behaviors.

With the integrated WISP framework, the final product will be a non-intrusive software application which meets the industrial needs of enhancing cyber security capabilities and can be easily adopted by utilities and RTO/ISOs.

8.3 Competitive Landscape

The team conducted a detailed survey on the existing commercial products addressing cyber security concerns at different levels of energy system operation. To the best of our knowledge, there is no product offering similar functions to WISP. This section provides a summary of a few relevant software and hardware solutions protecting the SCADA and energy management system (EMS) of bulk power systems. These can be complementary to WISP.

Lack of visibility is one of the main challenges for securing ICS. Operators often have more access to devices on the IT network than those on an ICS/OT (operational technology) network. The Dragos Platform [141] claims to provide a comprehensive security solution with integrated capability of asset visibility, threat detection and incident investigation to oversee large volumes of devices on the operational network. A Dragos case study report [142] described how the Dragos Platform was deployed on a mid-sized electric utility to protect the communications of its EMS. The Dragos Platform is a network monitoring software where all services are based on data mining of the network traffic. In contrast, WISP takes in the electricity market data as a reflection of system physical behavior without risking increased network overhead and interruption.

In response to the increasing cyber threats, major vendors/suppliers in the energy sector have started to develop their own cyber security features compatible with their existing products. One example is ABB's Network Manager SCADA/EMS/GMS [143]. As a main selling point, Network Manager claims to elevate utilities' security level to be compliant with certain CIP standards, via advanced security functions such as role based access control, 2-factor au-

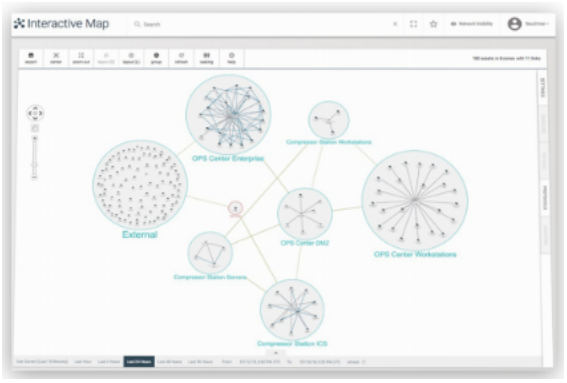


Figure 8.4: Dragos Platform Asset Visualization

thentication, and network encryption and ports lockdown. Another example is the SIEMENS RUGGEDCOM products [144], which covers a full range of industrial communication network devices from switches, routers, and media converters to servers. RUGGEDCOM provides built-in security appliances as well as integration capability to third-party solutions, such as CheckPoint and Secure-NOK. RUGGEDCOM also allows CIP enforcement and secure connections to Cloud services. One last example is EcoStruxure Cybersecurity Admin Expert [145] from Schneider Electric, a software tool ensuring configuration security to OT devices.

One main challenge for the deployment and marketing of cybersecurity products is the compatibility to legacy devices in operation. Security companies have to partner with the major electric equipment suppliers to be accepted by their customer base. For example, Dragos has support from GE, Emerson and SEL, while Secure-NOK is backed by SIEMENS. These partnerships are clearly a double-edge sword that helps with the initial market penetration but blocks further expansion. WISP breaks this barrier as an independent and non-intrusive solution that is not tied to the specifications of the physical/cyber infrastructure. WISP is only driven by publicly available information and is programmable to be adapted to different data formats. The innovation of WISP also lies in its unique detection mechanism that is based on system behavior rather than single device or network traffic behavior. To this extent, WISP can work together with the above listed solutions and provide additional situational awareness to operators.

Another observation is that the main motivation and driving factor for utilities/ISO/RTOs to invest on cybersecurity is the compliance and audition of CIP standards. This implies that solutions in the scope of CIP requirements are easier to market. As mentioned above, current CIP standards only partially fulfil the federal security guideline; novel security solutions are encouraged to help close the gap. WISP fits into the "Detect" element of the NIST framework. Unlike the cyber system anomaly detection required in CIP-007-6, WISP offers physical system anomaly detection which is equally important for securing highly coupled cyber-physical systems (CPS), especially electricity delivery systems. This unique feature makes WISP well-positioned for an unexplored market.

8.4 Path to Commercialization

8.4.1 Raytheon Technologies

Raytheon Technologies is one of the world's largest aerospace and defense companies. Raytheon Intelligence & Space (RIS), one of the four Raytheon Technologies business units, delivers full-spectrum cyber, training and service solutions to civil, military and commercial customers around the world. Especially, the RIS Cyber Physical Systems Security (CPSS) [146] team provides solutions to protect engineered sensing, control, computing and

networking systems in the critical infrastructure of smart cities, electricity grids, transportation and agriculture. Direct customers of CPSS include electricity utilities, industry regulation authorities/associations, federal agencies (DOD NAS) and military bases. These capabilities enable Raytheon Technologies to be in a strong position to commercialize WISP technology and promote it to existing and potential customers.

8.4.2 Deployment Plan

Independent software solution on premise or in cloud

The WISP software is an end-to-end cyber monitoring system with integrated functions from data downloaders to anomaly detectors and result visualizers. Especially, the WISP cyber-alert visualization board, shown in Figure 8.5, reports to the operators in real time from top to bottom: the number of anomalies detected, the top impacted nodes (locations), the raw locational marginal price (LMP), the corresponding anomaly score and the histograms. The operators can choose which node to monitor using the drop-down menu on the board or take input (recommended vulnerable nodes) from the vulnerability analysis module. Additionally, WISP provides an optional geographic contour map for a global view of the LMP values and anomaly scores, shown in Figure 14.8.

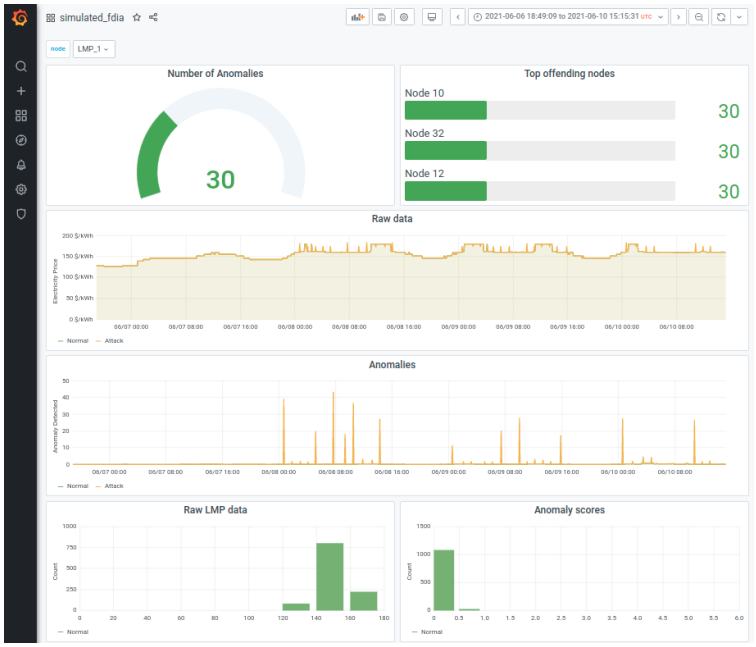


Figure 8.5: Current Version of WISP Visualization Board

WISP can be deployed as on-premise software on company-owned servers and behind the firewall. In this case, the maintenance of WISP will be offloaded to the customer with no recursive fees after purchase. WISP can also be deployed as an application in the cloud with

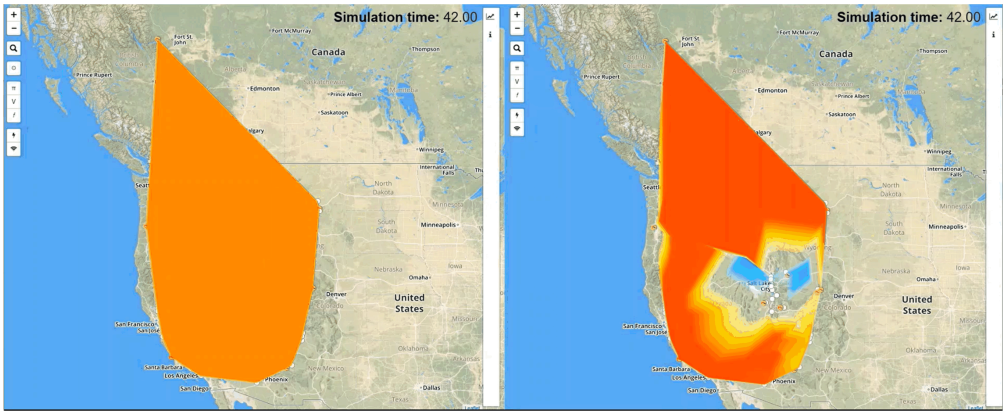


Figure 8.6: LMP contour map of California system with (right)/without (left) cyber attack

no hardware cost for computing, storage, and networking. The customer can avoid paying for extra redundancies to host WISP software and pay only for what is needed depending on the scale of the target system and complexity of the functions. The potential customers for WISP as standalone software are large utilities, ISO/RTOs, and industry regulation entities. However, they are not especially motivated to adopt WISP since it is not part of the CIP commitment. One way to promote this product is to offer a free trial license to set up an on-site demonstration and prove its value in field operations. Profits will be achieved when customers learn the benefits and become more confident to renew the license or upgrade the services.

Embedded detection engine in EMS/Cybersecurity software

Another path for commercialization is to market WISP as a backend cyber detection engine integrated with existing EMS or cybersecurity software. Major EMS systems already have powerful data management systems and operator view panels. Thus, WISP can be integrated into an EMS as a cybersecurity application that fetches data from historian and real-time databases and presents results to a panel with which operators are more familiar. Similarly, the cyber monitoring software, like the Dragos Platform and Secure-NOK, are also capable of data collection and topology visualization. This makes it easier for WISP to fit in and provide physical-behavior based cyber detection, complementary to their existing functions.

In these cases, the potential customers are EMS vendors or cybersecurity companies who seek to extend their systems and lead the frontier of cyber protection in the energy sector. Their end users, i.e. utilities/ISOs/RTOs, might be more comfortable to adopt the technology since it is offered as an add-on item to the existing solution.

Third-party information provider

Due to its non-intrusive nature, WISP can function without proprietary knowledge of the running power systems. The detection and diagnostic results of WISP can be compiled into analysis reports on a weekly, monthly or annual basis. These reports are valuable to utilities and government agencies that are generally concerned with the cybersecurity of

critical infrastructure. Though we all agree major blackouts severely interrupt people's daily life and industry production, early detection through physical symptoms are not achieved nowadays. We still heavily rely on the network intrusion detection systems which do not react to attacks that bypass the network interfaces, such as insider attacks, backdoor attacks or side-channel attacks. In cyber war, attackers could leverage multiple channels to craft more sophisticated and hidden intrusions. Intruders may be experts in concealing cyber traces, but they often overlook the physical traces. This makes the WISP report especially helpful to customers to gain additional situational awareness.

As an information provider, we identified a potential partner, GENSCAPE. One of their main business fields is to provide deep insights of the electricity market to prepare market participants with better trading strategies. WISP is not in their domain but partly shares the same mechanism (deep mining of the electricity market data) and customers. Hence, we anticipate a collaboration with GENSCAPE will help WISP gain more attention and interest from potential clients.

8.5 Customer Engagement and Outreach

ISO New England

To understand customers' perspectives and expectations, we visited ISO New England and presented our on-going project. The concept of WISP was well-received but with concerns of the software deployment overhead, detection scopes and false alarms. As discussed above, the computing resources and human labor needed for WISP operation and maintenance depends on the deployment plan. Since ISO New England has established a cybersecurity center, less effort is needed in training the staff to be cyber aware and understand the concept of WISP. We also clarified that WISP only detects attacks that leave physical traces or create disturbances in the electricity market data. Data sniffing or information leakage kind of attacks are out of scope for WISP. WISP introduced multiple advanced machine learning algorithms facilitated with domain knowledge to reduce the false alarms. We are aware that the false alarms draw unnecessary attention from the operators and thus we have kept the false alarm rate lower than 1% (about 3 per day) and plan to further reduce it to 0.1% (one per 3 days) in subsequent development.

PJM

The COVID-19 pandemic has largely obstructed our outreach activities. Instead of on-site visits, we organized an online meeting with the PJM technical team. During the meeting, the PJM team educated us on how the reserve capacity influences the real-time market and how PJM's market regulation rules shaped the prices. Inspired by their approach, we added the reserve market to the WISP simulator and added market rules for price spike detection. These functions enable WISP to handle uncertainties in the real-world data.

Industry Partners of UTK CURENT center

WISP is supported by the Center for Ultra-wide-area Resilient Electric Energy Transmission

Networks (CURENT) research center at the University of Tennessee. Their industry members include major power industry vendors (ABB, GE, Eaton, etc.) and utilities (Dominion Energy, ISO New England, PJM, etc.). We presented WISP at CURENT's summer retreat and annual industry conference to more than 100 audience and received valuable feedback from potential customers and partners. Our next step is to present a demo video at the next CURENT event to gain more visibility within the industry.

8.6 Conclusion

Modern electric grids face significant cyber risks due to the adoption of intelligent devices and systems. Along with the rapid increase of cyber-attacks, threats and vulnerabilities is the trend of evolving, advancing and persistent attacks targeting power systems. To protect the national electric grid, the U.S. government and industry regulation authorities enforced multiple cybersecurity standards as amendments to the existing NERC CIP. Asset owners and operators, together with the equipment vendors and cybersecurity specialists, have made significant progress to defend the grid against cyber criminals. However, they are mostly driven by the required commitment to the CIP standards and thus are limited by the scope of CIP. It is claimed in the GAO report that CIP only partially fulfils the cybersecurity guideline for critical infrastructure. By introducing WISP to the market, we hope to fill this gap and elevate our customers' security level to an even higher level so that they are better prepared for the upcoming surge of cyber intrusions. We identified three major paths for deployment and commercialization. Each leads to different customers, partners, costs and advertising activities. Raytheon Technologies, as a top cybersecurity service provider, is ready to explore these options. Further actions for the success of WISP commercialization include:

- Survey of the identified potential customers and partners to understand their interest and need
- Adopt the feedback in product design to fulfill customers' needs
- Reach out to government security agencies (NIST, NSA, DOE) to get support or certification to gain product credibility
- Open demonstrations and public presentations on various applications

Chapter 9

Conclusions - Phase I

The overall objective of WISP is to deliver a non-intrusive electricity market monitoring tool that detects and localizes the cyber-attacks using public-available locational electricity price data with other system information to provide additional cyber awareness to operators. To fulfill this, the major tasks include identify attack scenario and threat model; develop electricity market simulator and generate realistic datasets; develop and evaluate anomaly detection modules; system integration and software development. This report described the achievements in Phase I where each of the major tasks are addressed and explained. Specifically, we have developed anomaly detection algorithms including probabilistic detection models, deterministic detection models, ensemble models, locational detection models and price spike detection models. We have developed vulnerability and risk analysis module and root cause analysis module and completed the system integration in software development. The final WISP software is tested thoroughly following the product test plan and demonstrated in real-time. In summary, we have accomplished the major tasks in Phase I and we are well prepared for tasks in Phase II.

Chapter 10

Introduction - Phase II

The core focus of WISP is to observe publicly available prices to identify and explain the cause of anomalous pricing behaviors, either it is due to intentional or non-intentional acts on the underlying power system and market interfaces. In Phase 1, the team has (1) developed an electricity market simulator to generate cyber-attack data, (2) developed algorithms for anomaly detection, vulnerability analysis and root cause analysis, and (3) developed a software prototype for cyber monitoring. The Phase 1 software products were intensively tested through the IEEE 39-bus system. However, demonstrating the WISP software on large-scale systems is still a challenging task. In Phase 2, more efforts are required to build the use case interfaces, generate realistic testing datasets, optimize the software performances, refine the algorithms and redesign the result presentations.

Phase 2 of WISP is executed as follows. The team started with red team testing to identify cyber vulnerabilities of the WISP software and implement the mitigation approaches to satisfy cyber risk management requirements. In order to generate realistic cyber-attacks for large-scale systems, the team improved the electricity market simulator by upgrading the optimization solver and adopting DC power flow in attack algorithms. Furthermore, the software prototype was modified to enable efficient and robust operation on large systems. The database was restructured for faster data flow. The detection was programmed to be parallel computing with data buffers allowing for less hard-disk data queries. The visualization platforms were integrated to provide timely and comprehensive result presentation. The detection models were retrained and the parameters were fine-tuned to achieve the best detection performance. The demonstration details were elaborated in this report.

The remainder of this report is organized as follows. Chapter 11 presents the red team testing procedure and observations. Chapter 12 presents the electricity market simulator improvement and the attack scenario design. Chapter 13 presents the software improvement to accommodate the need of processing large-scale power systems. Chapter 14 presents the demonstration results and analysis for the Texas system and the ISO New England system. Chapter 15 concludes Phase II.

Chapter 11

Red Team Testing

The Raytheon Technologies Research Center cybersecurity team performed a red team engagement at the request of the WISP development team to identify the full range of realistic threats and their impacts to the WISP software. The Red Team identified several exploitable vulnerabilities and leveraged them to perform attacks that compromised information confidentiality, data integrity and system accessibility. The team also provided recommendations for security hardening solutions to mitigate these impacts. The red team activities are summarized in this Chapter.

11.1 Objective

The overall objective of the red team testing is to identify potential vulnerabilities via real-world adversary techniques and provide corresponding mitigation recommendations to the system under test. The red team activities should support the following objectives:

1. To define the test tools and environment needed to conduct the test.
2. To define the sources of the information used to conduct the test.
3. To perform vulnerability analysis and cyber-attacks for the system under test.
4. To communicate testing results and mitigation solutions to the development team.

11.1.1 Background

The Raytheon Technologies Research Center (RTRC), in collaboration with the University of Tennessee and the Pacific Northwest National Laboratory, developed a cyber-monitoring software, called WISP (Watching grid Infrastructure Stealthily through Proxies), under the sponsorship of the U.S. Department of Energy, CEDS program. The project started at March 25th 2019 and the Phase I Research and Development was completed at May 30th 2021. As a result of Phase I, WISP was prototyped as an end-to-end software that is able to be interfaced with major electricity market data and/or simulation data to train and test the anomaly detection core and print the results on a visualization panel.

As a cyber-monitoring tool, WISP could be deployed in the same environment with the system under protection. For example, WISP can be embedded in the Energy Management System (EMS) of a utility or ISO. Therefore, the security assurance of the WISP software itself is a prerequisite for customer adoption. The WISP development team invited the RTRC cybersecurity team to perform a thorough red team testing aiming to minimize the security risks.

11.1.2 Scope

The red team testing targets on all interfaces of the WISP software. It does not include vulnerabilities related to the customer's security policies and procedures. The testers assume no constraints of the attackers and assume to have access to the network that hosts the WIPS software. Thus, the testers could focus on the adversary techniques specific to the WISP software instead of the IT-only techniques.

11.2 Referenced Documents

WISP application should provide a layered and resilient security protection of its systems based on preventative, detective, and corrective security controls from the NIST SP 800-53 (Rev4). The NIST 800-53 Security and Privacy Controls for Federal Information Systems and Organizations (Revision 4) controls protect WISP application against threats and vulnerabilities with an acceptable level of risk. They protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation throughout the life cycle. These include general protections such as authentication, access controls, host and network intrusion detection, malware detection/protection, and firewalls as well as the appropriate use of cryptographic techniques for data protection and integrity.

The WISP security architecture should allocate security controls across the WISP application. These controls help recover WISP systems from security threats and are tied to appropriate Risk Management Framework (RMF) families to minimize challenges of RMF Assessment and Authorization.

The WISP Red Team selected NIST RMF Access Control, Systems and Communications Security, and System and Information Integrity families as the most important areas with cybersecurity concerns.

- **NIST RMF Access Control:** Identity Management, Authentication and Access Control
  Access to physical and logical assets and associated facilities is limited to authorized users, processes, and devices, and is managed consistent with the assessed risk of unauthorized access to authorized activities and transactions.

- **NIST RMF Systems and Communications Security:** Denial of Service Protection
  The information system protects against or limits the effects of the following types of denial-of-service attacks: volume-based attacks, protocol attacks, and application layer attacks by employing IDS, IPS, and firewalls.

- **NIST RMF Information Integrity:** Malicious Code Protection
  The organization employs malicious code protection mechanisms at information system entry and exit points to detect and eradicate malicious code; updates malicious code protection mechanisms whenever new releases are available in accordance with organizational configuration management policy and procedures; configures malicious code protection mechanisms to perform periodic scans of the information system and real-time scans of files from external sources at network entry/exit points as the files are downloaded, opened, or executed in accordance with organizational security policy; and block malicious code, quarantine malicious code, send alert to administrator in response to malicious code detection, and addresses the receipt of false positives during malicious code detection and eradication and the resulting potential impact on the availability of the information system.

The Red Team analysis contributes to the WISP System Security Plan (SSP) which describes the security requirements for the WISP system and the components that reside within this system's security boundary. The SSP supports controls allocated to the hardware and software components that comprise the WISP system. WISP SSP should be a living document that is maintained over time and will be configuration controlled to allow for the continuous monitoring process to evolve. It is crucially important that the document is maintained as a security control as system status changes over time.

11.3 System Examined

The Red Team evaluated the WISP software installed on a server with Ubuntu system in RTRC's Cyber-Physical Security Research Lab. WISP consists of three modules: the data management module, the anomaly detection module, and the visualization module. The data management module is based on a standard SQL database, called MariaDB. WISP provides data downloading, data formatting, data query and data clean functions in the data management module. The anomaly detection module supports offline training, model saving and loading, and online detection pipeline. The data are fed from the data management module with formats fit into each detector. The visualization model is developed on the open-source real-time visualization platform, called Grafana. Grafana provides customized visualization dashboard which is configurable to adapt to different data sources and data view requirements.

The WISP software potentially exposes two interfaces to the external and internal entities. One is the data downloading scripts that request real time electricity price updates for remote utility databases. The other one is the visualization web server that allows remote clients to view the detection results and raw data curves through authorized user accounts. The Red Team explores these two interfaces and searches for other unexpected open interfaces as attack entry points.

11.4 Red Team Approaches

Overall approach: the team used Kali Linux and the penetration testing software tools on Kali platform. The Kali machine is on the same network with WISP software host machine. The testing assumes all malicious activities have intruded into the enterprise network and the implementation describes the exploits of the WISP software exposure to an unauthorized entity on the same network.

Detailed implementation for each attack:

(1) SQL Attack

Attack Impact:

An attacker that can connect to the database can tamper with the raw data, the anomaly scores, and the user permissions. As a result, the system can become inaccessible, and the data can be untrustworthy.

Attack Implementation:

1. Kali/Nmap scanning was conducted to discover hosts and services on a computer network by sending packets and analyzing the responses. As shown in Figure 11.1, an open port 3306 was discovered, which is a reserved port for SQL services.
2. Kali/Hydra is a parallelized network login cracker built in various operating systems like Kali Linux, and other major penetration testing environments. Hydra uses different brute-force approaches to guess the right username and password combination. The team first tried several base key words using the software engineers' names and the software server's name. A base dictionary red_team_keywords.txt was created with keywords: Mark, Fragki, Lynn, Wisp2.
3. Hydra applied red_team_keywords.txt without success (Figure 11.2).
4. Next, the team used Hashcat to mangle the given red_team_keywords.txt and created an expanded wordlist, named red_team_mangleuniq.txt. Using this dictionary, Hydra was able to reveal the username and password combination for the SQL root user, as shown in Figure 11.3.

Mitigation: The SQL server should be configured to listen only to local addresses, and a strong password should be used that does not contain known and easily predictable words.

(2) SQL DOS

Attack Impact:

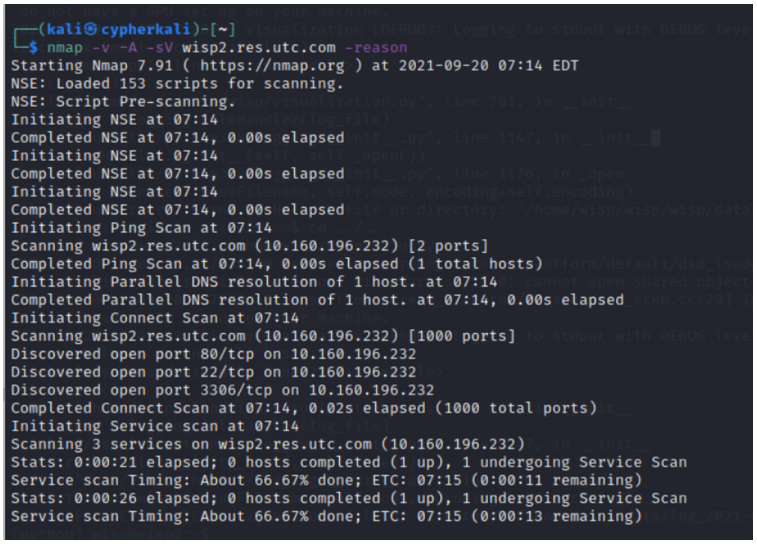


Figure 11.1: Nmap scanning results.

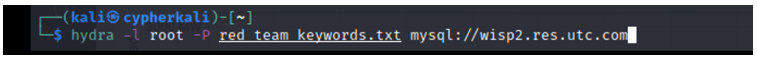


Figure 11.2: Hydra applied red_team_keywords.txt.

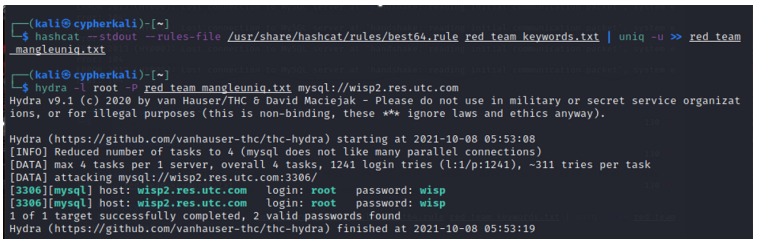


Figure 11.3: Hydra used red_team_mangleuniq.txt and cracked credentials of the SQL root user.

An adversary can render the system inaccessible by manipulating the rate of the queries to the database.
Attack Implementation:

1. The team logged into SQL with the cracked password and selected 'single_day' database for DOS attack, shown in Figure 11.4.
2. The team then launched dos.sh script to interrupt SQL services shown in Figure 11.5.

Mitigation: Appropriate quality-of-service (QoS) values should be set in the SQL server. This will block a single connection from consuming all server's resources and make it responsive to other clients' requests.

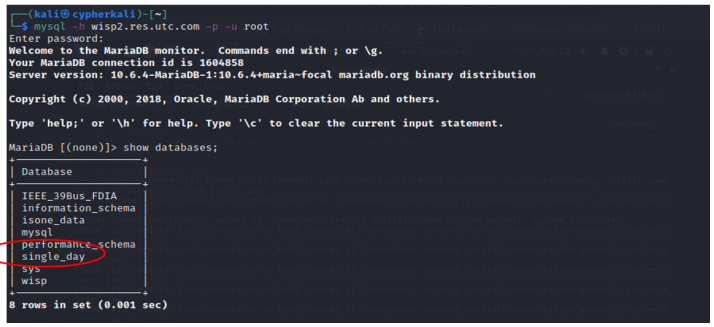


Figure 11.4: Login with the cracked password and select 'single_day' database for DOS attack.

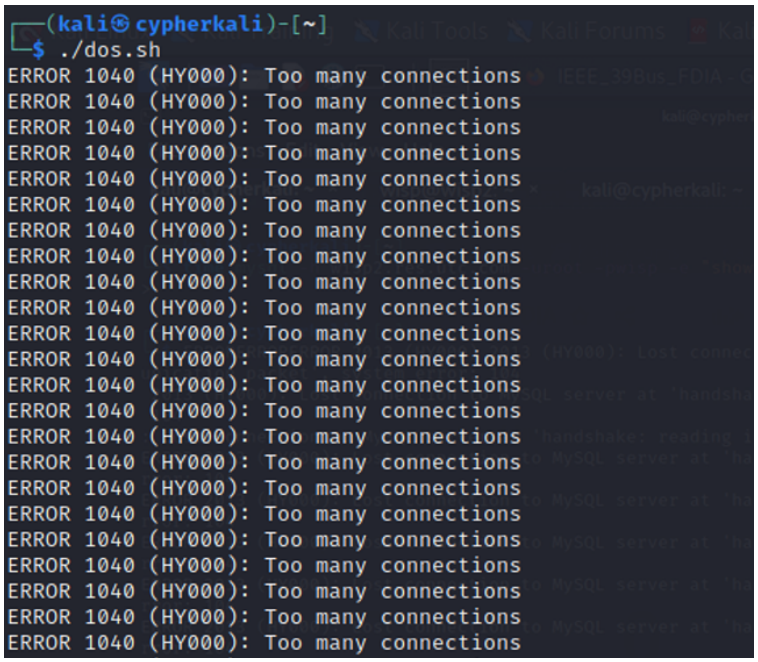


Figure 11.5: Apply dos.sh script to cause SQL DOS attack.

(3) Grafana Password Attack

Attack Impact:

An adversary that can authenticate in the front-end can pivot to other attacks and tamper with the visualization. Additionally, sensitive information can be leaked.

Attack Implementation:

1. The team used Kali/Hydra to brute-force Grafana's user credentials. A list of key words was created using some popular default values. A base dictionary wisp_keywords.txt

was created with keywords: wisp, test, admin, root.

2. Hydra applied wisp_keyword.txt with success (Figure 11.6).

Mitigation: Grafana should be configured to listen only to white-listed addresses, and a strong password should be used that does not contain known and easily predictable words.

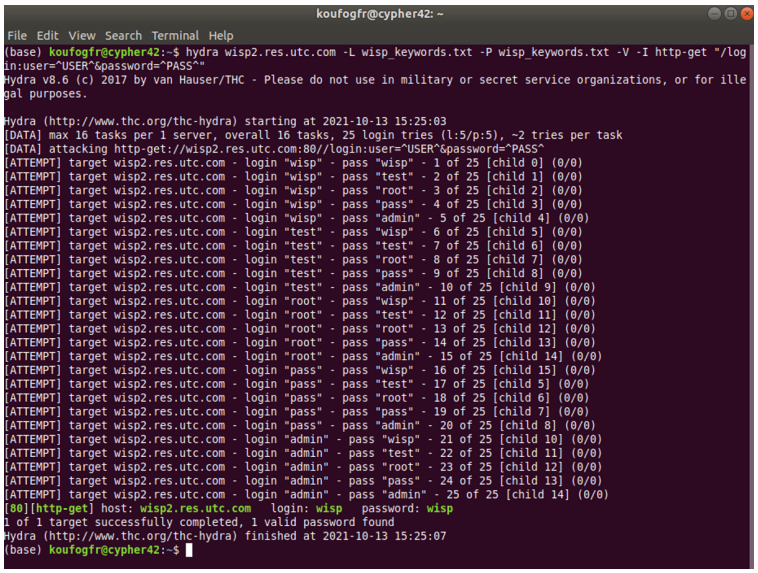


Figure 11.6: Hydra applied wisp_keywords.txt with success.

(4) SQL code injection from Grafana

Attack Impact:

An adversary can change raw SQL queries and make changes to the database through Grafana interface.

Attack Implementation:

1. The team has authorized access to Grafana using previously cracked credential. Figure 11.7 shows that Grafana is using a highly privileged account to access the database.
2. The team edited an existing dashboard and modified the SQL query to include malicious code (Figure 11.8).
3. When another user reloaded the dashboard, and the malicious SQL query was executed, and the attack deleted a whole table "anomaly scores" from the database (Figure 11.9).

Mitigation: Grafana should use a read-only account to access the database.

(5) Code injection to Grafana's dashboard configuration

Attack Impact:

An adversary can edit Grafana's dashboards and save them back to the server to manipulate the information visualized to other users. For example, an attack can hide detected anomalies.

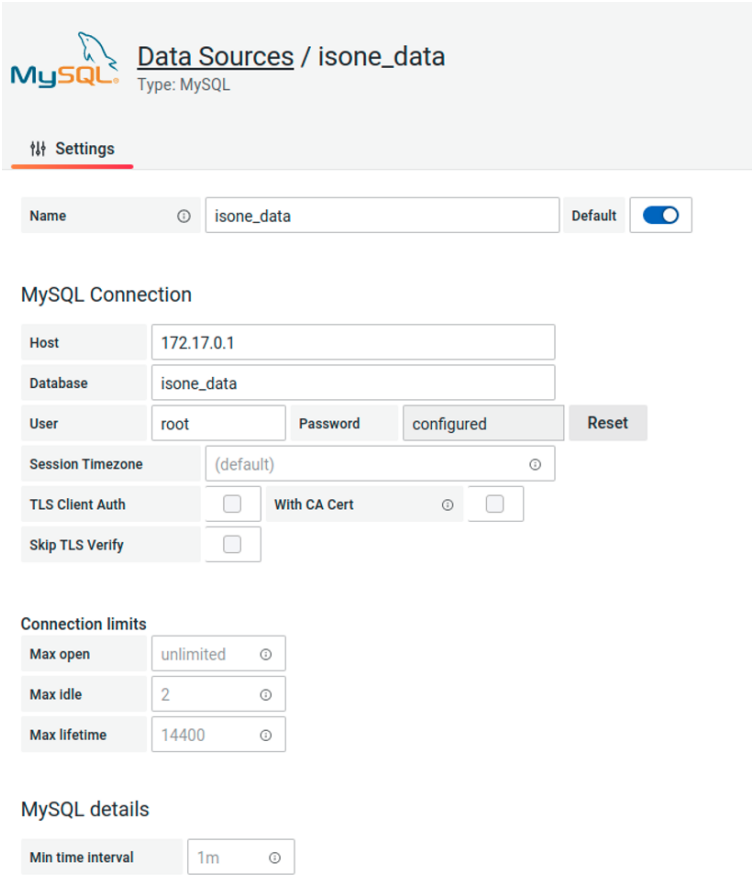


Figure 11.7: Grafana SQL is configured with root credential.

Attack Implementation:

1. The team logged into Grafana using previously cracked credential and selected a dashboard named "IEEE_39Bus_FDIA".
2. The team edited the dashboard and modified the SQL query commands to change the data retrieved from the database. (Figure 11.10)
3. The dashboard was saved back to the server, and a benign user inspecting the dashboard saw the modified data that contain incorrect load data. (Figure 11.11)

Mitigation: A Grafana user should have read-only permissions to the dashboards.

(6) Grafana SYN Flood DOS Attack

Attack Impact:

An attacker that can deplete the resources of the system can make the system inaccessible for a benign user.

Attack Implementation:

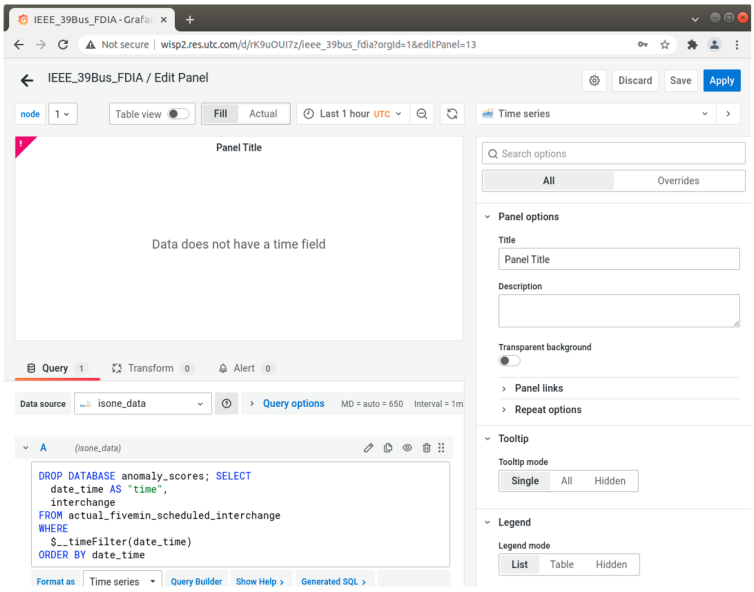


Figure 11.8: The team edited an existing dashboard and modified the SQL query to include malicious code.

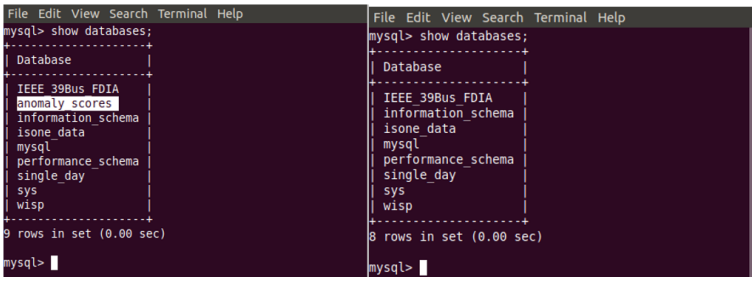


Figure 11.9: SQL has an "anomaly scores" table (left) vs. Malicious code deletes the "anomaly scores" table (right).

1. To simulate DOS attack, the team used Kali/HPING, an open-source packet generator and analyzer for the TCP/IP protocol. HPING is often used for security auditing and testing of firewalls and networks. In this attack, the team used HPING tool to send SYN (synchronization) messages to Grafana in flood mode, which exhausted Grafana's SYN-ACK (synchronization acknowledgement) connections to other users, as shown in Figure 11.12.
2. As a result, Grafana cannot be loaded for other users, as shown in Figure 11.13.

Mitigation: Grafana should be deployed within a monitored network with appropriate firewall rules that detect and block the most common types of network attacks.

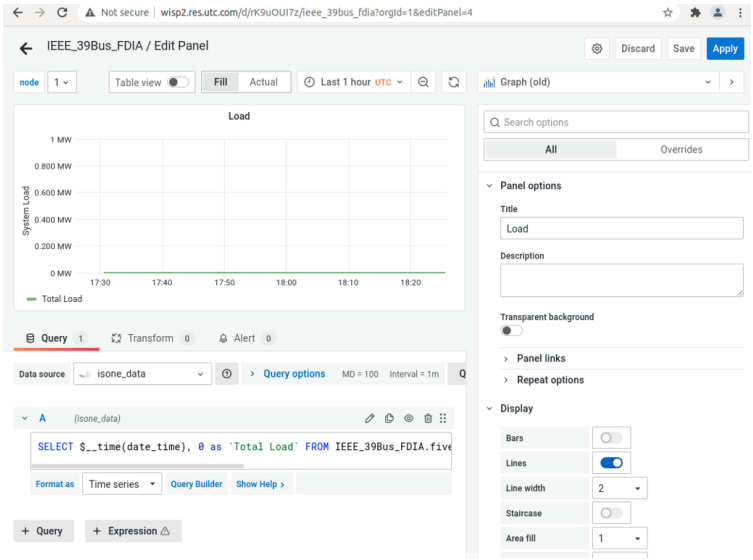


Figure 11.10: The team modified the SQL query to replace all data retrieved from the database with zeros.

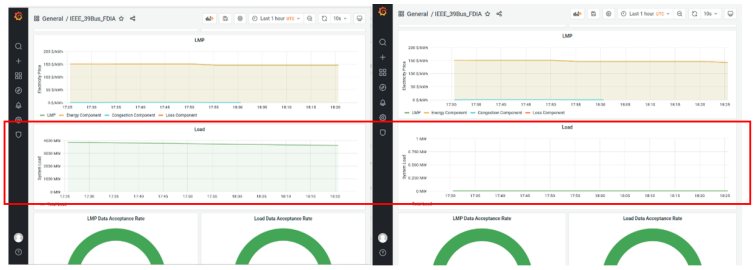


Figure 11.11: Grafana dashboard shows benign data (left) vs. Grafana dashboard shows attack data (right).

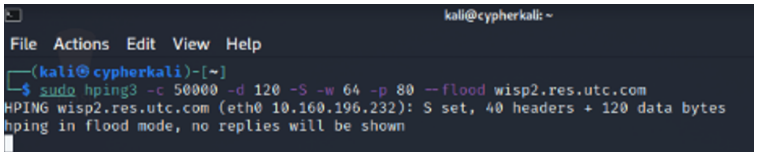


Figure 11.12: HPING tool sent echo request to Grafana in flood mode.

11.5 Result Analysis

The system vulnerabilities identified through red team process are listed in this section. We also provide potential mitigation methods to be compliant with the NIST framework.

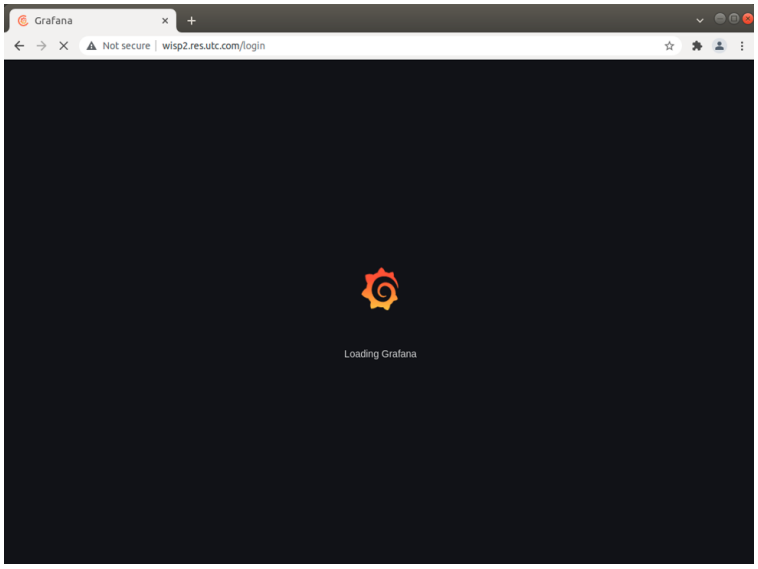


Figure 11.13: Grafana cannot be loaded for other users.

11.5.1 Vulnerability

(1) SQL interface vulnerability:

- An attacker that can connect to the database can tamper with the raw data, the anomaly scores, and the user permissions. As a result, the system can become inaccessible and fake data can be visualized.
- An adversary that can control the rate of the queries to the database can render the system inaccessible. Grafana interface vulnerability:
- An adversary that can authenticate in the front-end can pivot to other attacks and tamper with the visualization. Additionally, sensitive information can be leaked.
- Several fields in Grafana access code that is passed to other components. Although Grafana already sanitizes this input, it allows for raw SQL queries to be sent to the database. An adversary can change these queries and make changes to the database.
- An adversary can edit Grafana's dashboards and save them back to the server, effectively hiding detected anomalies or publishing misleading information to other users.
- An attacker that can deplete the resources of the system can make the system inaccessible for a benign user.

(2) Grafana online vulnerabilities:

The team noticed that there are multiple CVEs against Grafana v8.1.2, a version currently used in WISP, shown in Figure 11.14.

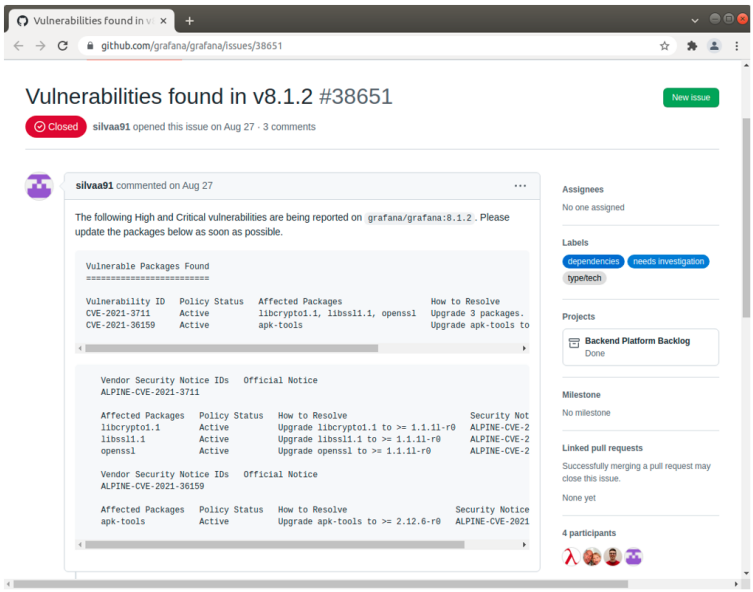


Figure 11.14: Grafana 8.1 has known vulnerabilities.

11.5.2 Mitigation

(1) SQL interface hardening:

1. The SQL server should be configured to listen only to local addresses. If needed for development, a separate account should be created and the root account should not be exposed to external addresses.
2. A strong password should be used that does not contain known and easily predictable words.
3. A cooldown period after a few unsuccessful login tries can slow down an attacker's brute-force effort.
4. Appropriate quality-of-service (QoS) values should be set in the SQL server. This will block a single connection from consuming all resources of the server and make it responsive to other clients' requests.

(2) Grafana interface hardening:

1. A strong password should be used that does not contain known and easily predictable words.
2. Grafana should be configured to listen only to white-listed addresses. If access over the Internet is needed, a VPN or a DMZ should be used.
3. Grafana should use a read-only account to access the database.
4. A Grafana user should have read-only permissions to the dashboards. Therefore, any changes made cannot affect other users' view.

5. Grafana should be deployed within a monitored network with appropriate firewall rules that detect and block the most common types of network attacks.
6. Grafana should also be configured to automatically update to the latest version.

11.6 Conclusions

The RTRC Red Team performed a series of adversary attacks to determine the security level of the WISP software. These attacks target mostly on the data management module and the visualization module. The major observations include (1) the SQL database was exposed to all entities on the network; (2) the SQL database was not configured to enforce strong password policies including limitations on password retries; (3) the Grafana platform also does not enforce strong password policies; and (4) the Grafana platform was not configured to limit the user privilege to the database access and dashboard editing. Besides several recommendations on secure configuration, we also recommend adding new firewall rules to limit the accessibility of WISP remote users.

Chapter 12

Electricity Market Simulator and Attack Scenario Design

In Phase 2, we tested and improved the electricity market simulator for a large-scale power system use case, i.e. the Texas synthetic 2000 bus system. To create realistic and impactful cyber-attacks, we searched for the critical power components and vulnerable time periods for the attack scenario design. In this chapter, we first introduce the structure of the electricity market simulator, following by the improvement for large-scale system simulation. We then elaborate the attack scenario design for the final demonstration.

12.1 Electricity Market Simulator

To generate realistic electricity market data with cyber-attacks, we developed an electricity market simulator in Phase 1. In this section, we first review the structure and components of the simulator. Since the simulator was only tested on IEEE 39-bus system in Phase 1, we met a few challenges when testing it on a large system. This section explains the challenges and the improvements for the simulator to properly generate data for Texas 2000-bus system.

12.1.1 Structure of electricity market simulator

The electricity market simulator is designed to take in real-time five minutes load data to generate locational marginal prices and relevant physical measurements (e.g. power flow data and state estimation data), provided the topology and parameters of the system and configurations of the cyber/physical contingencies. The overall structure of the simulator is illustrated in Figure 12.1.

The simulator consists of two simulation loops: the main loop and the reserve loop. In the main loop, the simulator starts by computing the DC optimal power flow (DCOPF) based on the current load level and system status. The results are then used to generate AC measurements with perturbation of the measurement noises. The simulator can then

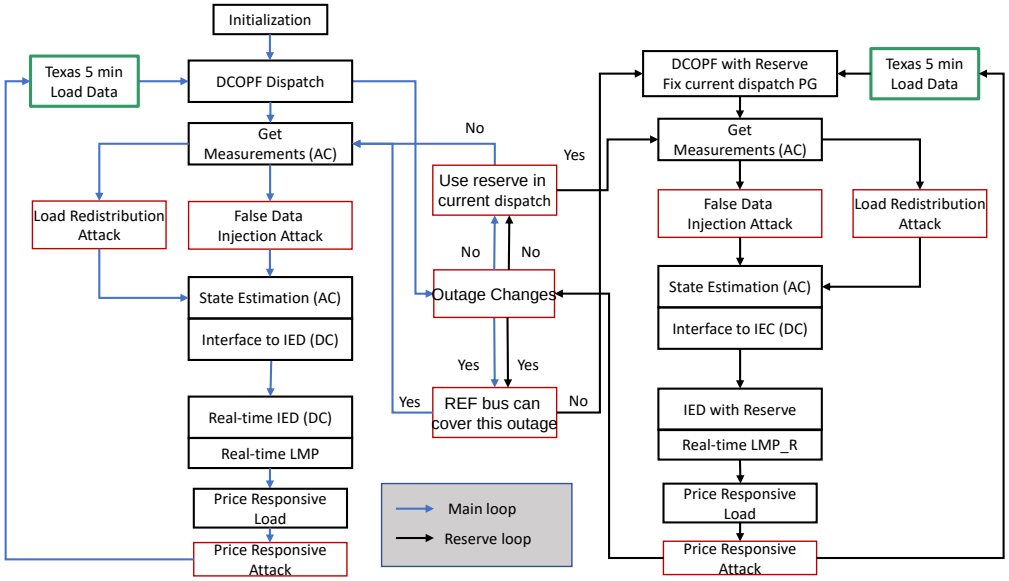


Figure 12.1: Structure of electricity market simulator.

inject false data injection attacks or load redistribution attacks in the measurement data. The processed data is used to compute the state estimation. The estimated states will be converted to DC power flow to prepare for the incremental economic dispatch (IED), which produces the real-time locational marginal prices (LMPs) and the final dispatch results. If the system is configured to have demand response program, the incremental load will be computed and the price responsive attack may be applied at this step. The adjusted load and the next time step load profile will be used for the subsequent loop. The pipeline of the reserve loop is similar with the main loop except that it adds the reserve capacity to the total generator capacities when computing DCOPF.

The switch between the main loop and the reserve loop is controlled by an outage scheduler and a system recovery flag. The outage scheduler defines when, where and what type of outages occur in the system. The system recovery flag shows if the system has recovered and if there is a need to stay in the reserve loop. Specifically, after each DCOPF in the main loop, the simulator will check if there is an outage status change. If outage status changes (turn ON or OFF), the system will check if the reference bus (REF bus) can cover the capacity gap caused by the outage event. If yes, the system stays in the main loop, otherwise, it switches to the reserve loop. Similar status check happens after each last step of the reserve loop.

On the other hand, if the outage status stays the same (no status change), the simulator will check if the system is currently using reserve. If yes, it means the system still needs to

stay in or switch to the reserve loop. If not, it means the system has recovered from the outage or there is no impact from any outages at this time step, it can stay in or move back to the main loop. The Phase 1 report provides the details of the algorithms and functionalities of each component in the simulator.

12.1.2 Large-scale system simulation

The main challenge for simulating the electricity market of a large-scale system is the convergence and computing speed of the AC optimization solvers. AC optimization is used in the process of false data injection attack generation and state estimation. AC power flow is more accurate in representing the real-world physical conditions than DC power flow which ignores the reactive power and line conductance. However, AC power flow optimization is an non-linear non-convex problem that often leads to loss of convergence or very slow convergence. To address this issue, we first upgraded our optimization solver from Matlab *fmincon* solver to Gurobi optimizer (9.1.2). This update guarantees the convergence of the large-scale Texas system, but the computing speed is still very slow. We then removed the AC conversion and kept only the DC power flow for the false data injection attack and state estimation. This modification largely reduced the simulation time and allowed us to search for the best attack period and locations exhaustively for the whole system and for a whole year data profile.

12.2 Attack Scenario Design

Unlike load redistribution attacks (LRAs) and price responsive attacks (PRAs), false data injection attacks (FDIAs) are more researched in the literature and widely accepted to be both realistic and impactful. For Texas 2000-bus system, the LRA and PRA are tested to be non-impactful given realistic parameter settings. Thus, in Phase 2, we only focus on the study and data generation of the FDIAs. In this section, we will first review the mechanism of FDIAs and then introduce the process and results of the attack scenario design for the Texas system.

12.2.1 False Data Injection Attacks

FDIA needs to be well-designed to bypass the bad data detection (BDD) examining, and hence it is actively researched, both for designing a successful realistic attack and for finding the defense countermeasure to protect the power system [29,56,65,68,69]. Based on the most practical state estimator and BDD scheme, let $\vec{a}$, $\vec{z}_a = \vec{z} + \vec{a}$ and $\vec{\hat{z}}_a$ denote the false data injection vector, fake measurements and state estimation results from the fake measurement, respectively. Without carefully constructing the malicious data $\vec{a}$, the residual $\vec{r}_a = \vec{z}_a - \vec{\hat{z}}_a$ can break the residual test and hence be easily detected by BDD.

In order to successfully hide the malicious attack, the attack vector $\vec{a}$ must satisfy the condition

$$\vec{a} = h(\vec{\hat{x}}_a) - h(\vec{\hat{x}}), \tag{12.1}$$

where $\vec{\hat{x}}_a$ is the estimated state under FDIA. In order to construct $\vec{a}$ satisfying (12.1), we follow a similar strategy proposed by [56], to minimize the changes in the states while lunching a successful attack. We can formulate this optimization as

$$\begin{aligned}
\min_{\Delta\vec{V},\Delta\vec{\theta}} \quad & \|\Delta\vec{V}\|_2^2 + \|\Delta\vec{\theta}\|_2^2 && \text{(12.2a)}\\
s.t \quad & P_i^{inj}(\vec{V},\vec{\theta}) = P_i^{inj}(\vec{V}+\Delta\vec{V},\vec{\theta}+\Delta\vec{\theta}), \forall i \in \mathbb{B} && \text{(12.2b)}\\
& F_{target}(\vec{V}+\Delta\vec{V},\vec{\theta}+\Delta\vec{\theta}) \geq F_{target}^{max}, && \text{(12.2c)}\\
& \Delta V_i^{min} \leq \Delta V_i \leq \Delta V_i^{max}, \forall i \in \mathbb{B} && \text{(12.2d)}\\
& \Delta\theta_i^{min} \leq \Delta\theta_i \leq \Delta\theta_i^{max}, \forall i \in \mathbb{B} && \text{(12.2e)}
\end{aligned}$$

where

$\Delta\vec{V}$: changes happened to the bus voltage;
$\Delta\vec{\theta}$: changes happened to the bus phase angle;
$\Delta\theta_i^{max}$: maximum changes in phase angle at bus i;
$\Delta\theta_i^{min}$: minimum changes in phase angle at line i;
ΔV_i^{max}: maximum changes in voltage magnitude at bus i;
ΔV_i^{min}: minimum changes in voltage magnitude at bus i;
$F_{target}(.)$: the real power flow on the targeted line;
$P_i^{inj}(.)$: the power injection at bus i.

The idea of this optimization problem is to find the minimum changes to the states, subject to (1) keeping the same power injection at all the buses, and (2) creating congestion at the targeted line. It is worth mentioning that this optimization problem is hard to solve, and there is no guarantee to find a solution. This is due to the fact that if the flow on the targeted line is far away from its limit, there is no such solution that can make this line congested while keeping all the power injections unchanged. Therefore, we only apply this attack when the flow on the target line is close to its limit. Solving this problem gives us the attack vectors to the state $c = (\Delta\vec{\hat{V}}, \Delta\vec{\hat{\theta}})$. We can then get the full attack vector by setting $\vec{a} = h(\vec{\hat{x}} + \vec{c}) - h(\vec{\hat{x}})$. This FDIA can successfully push the target line flow to the limit and make it look congested in the state estimate. Therefore, the following IED, which uses this fake state estimation, cannot assign any more flow to the target line, and has to shift the flow onto other routes if needed. More detailed discussion and implementation of FDIAs can be found in the Phase 1 report.

12.2.2 Attack Scenario Design for Texas System

The Texas 2000-bus system contains 2000 buses (542 generator buses) and 3206 branches (lines). For the one year load data profile, we first selected the peak load period (June, July and August) when the system is stressed to supply the demand. During the selected time period, we then searched for the congested lines or nearly congested lines. For these lines, we apply FDIAs with a timing condition constrained by the load status and the congestion level of the target line. One observation is that the peak load time of a day may not be the best attacking time due to the fact that the system is already prepared for the high demand. Study [118] shows the most vulnerable time to attack is at the critical load levels when even small incremental loads can cause major price changes by triggering expensive marginal units. We leveraged this observation by adding a searching condition to compare the increasing/decreasing delta load between two time steps. If it is above the average level, we will apply the FDIA and measure the price impact. There is a higher chance of creating impactful attacks using this strategy. The searching results for line 805 are shown in Figure 12.2.

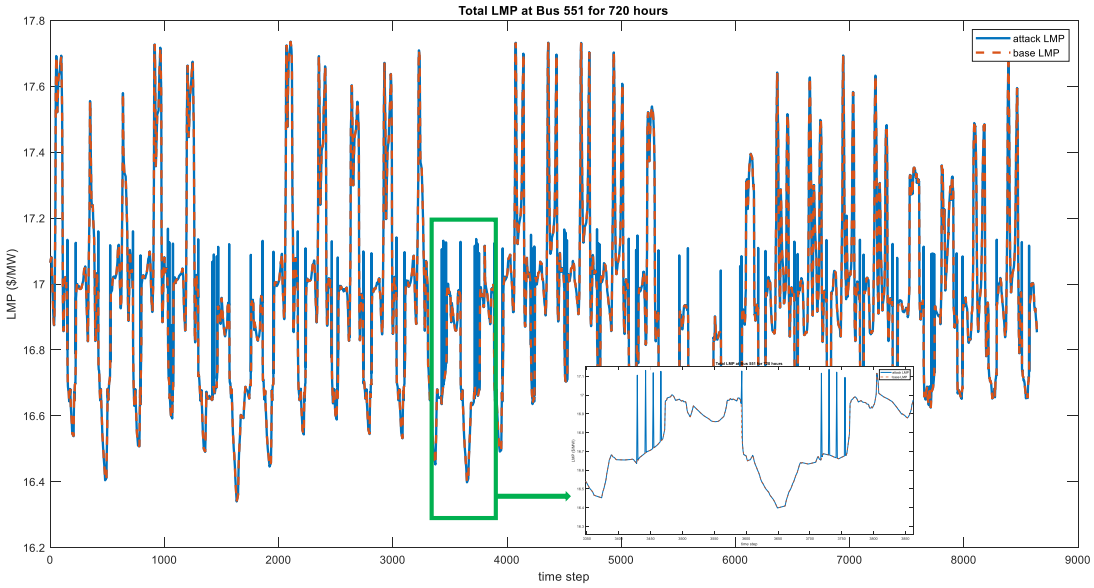


Figure 12.2: Baseline LMP and attack LMP at bus 551 for 720 hours under FDIA at line 805.

From the searching results, we can see the most impactful attacks were applied at the load transition periods rather than the peak load periods, which is consistent with our observation.

12.3 Conclusions

This chapter presents results from the effort of improving the electricity market simulator and design cyber attacks for large-scale power systems. The optimization solver was upgraded and the attack model was restricted to DC models to guarantee convergence of the simulator. The Texas 2000-bus system was studied for the most impactful FDIAs and the results of a successful attack scenario was presented and selected for final demonstration.

Chapter 13

WISP Software Improvement

WISP is an energy market monitoring tool that uses public energy market data to detect potential cyber-attacks on the system. Real time and simulated market data is analyzed by various diagnostic algorithms and the resulting detection data and events are stored in a database to be displayed in a user friendly interface. In this chapter, we first review the framework of the WISP software, developed in Phase 1. We then elaborate the improvement of the software to adapt to the performance requirements of monitoring a large-scale power system.

13.1 System Overview

The framework of the WISP software is illustrated in Figure 13.1. The software consists of three modules: the data plane, the algorithms and the visualization.

The data plane offers a database manager which is a class of functions to create databases and provide utilities for data downloading, saving, fetching and formatting. The data plane interfaces to three data sources: the real-time data from PJM and ISO-NE, the historical data from PJM and ISO-NE, and the benign/attack data from the electricity market simulator. All data are stored in a MariaDB based database system hosted in a docker container, which is managed also by the data plane.

The algorithms provide three categories of data-driven anomaly detectors: the autoencoder detector, the pointwise anomaly detector and the system anomaly detector. The details of the detectors can be found in the Phase 1 report. The algorithms also provide two ancillary functions: the parameter tuning and threshold selection. Both of them are important tools for maintaining high level detection performance for different systems.

The visualization module is based on Grafana software which is hosted in a docker container. Functions in the visualization module are designed to initiate and configure the Grafana server and to generate dashboards for each monitored system.

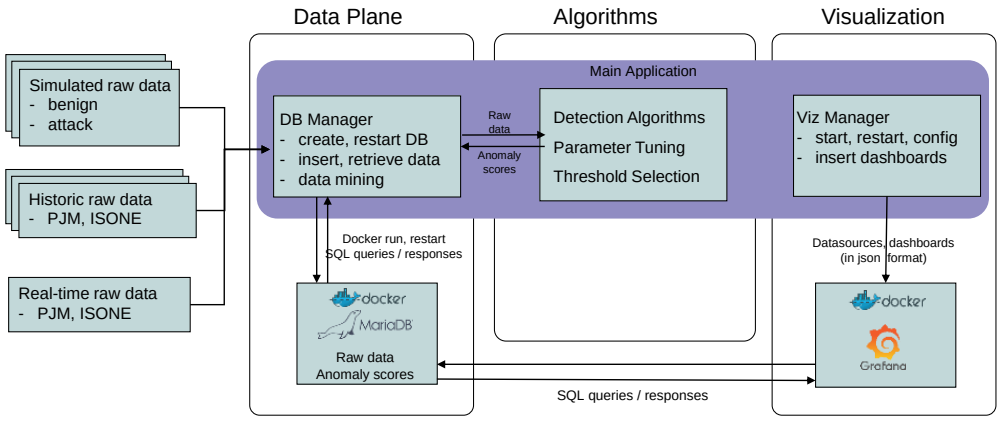


Figure 13.1: WISP Software Framework.

13.2 Software Improvement

In this section, we explain the challenges of using the Phase 1 software for large-scale systems and the solutions we developed to address these challenges.

13.2.1 Database Restructure

In the Phase 1 software, we used single timestamp index for the simulated data (IEEE 39-bus system). It works well because there are only 39 nodes (buses) in the system and data query by timestamp is sufficient and fast. When working on the Texas system, the data query became extremely slow since there were 2000 buses. If the location information is not filtered in the query stage, the post processing will become very slow and memory-consuming. For data efficiency, we changed all databases to composite index system with both timestamp and location as primary searching keys. This data restructuring helps reduce the data query time and also reduce the data size needed for each computing.

13.2.2 Computing Speed Optimization

To understand the end-to-end computing delay, we applied a Python profiler to record and rank the computing time for each function. An example of the profiler results is shown in Figure 13.2.

The three most time-consuming functions are data query, anomaly detection and data

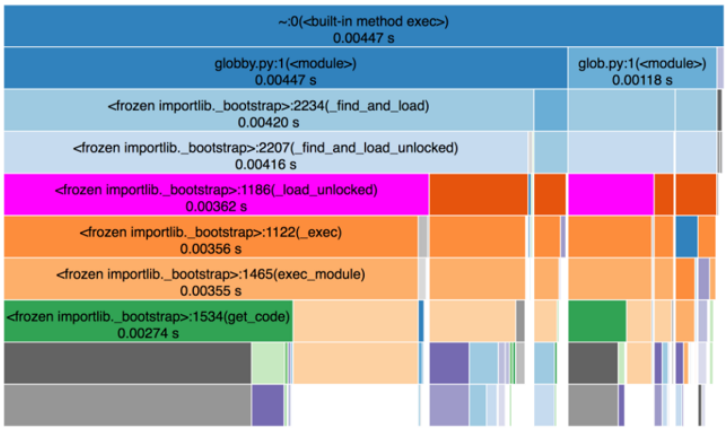


Figure 13.2: An example of software profiler results.

saving. For Phase 1 software, we used the single-thread serial computing structure. All detectors are executed in a predefined one-after-another sequence. The total computing time for each time step will be a sum of computing time of each detector. In Phase 2, we adopted the multi-thread parallel computing structure, where the software automatically launches an individual thread for each detector. The total computing time is the slowest computing time among all detectors instead of the sum of all. To further reduce the data query and data saving time, we introduced two global data buffers. The first buffer stores the data that are recently fetched from the system, so that the algorithms that share the same data input can visit this data buffer without starting a new database query. The second buffer stores the recent generated anomaly scores. This is used for stateful score computing where historical scores are needed to decide the final detection decision of the current time step. With this buffer, the algorithms do not need to visit the database for previous scores which largely reduced the hard-disk visiting time.

13.2.3 Ancillary Function Automation

In Phase 1, we provided the hyper-parameter tuning in a grid-searching strategy. The algorithm will test all possible combinations of parameters and find the best set. However, this process is very slow and a manual configuration is needed for more refined searching. In Phase 2, we leveraged the Bayesian optimization based parameter searching to automate and accelerate the tuning process. Bayesian optimization works by building a surrogate function (in the form of a probability model) of the objective function $P(score|hyperparameters)$. The surrogate function is much cheaper to evaluate than the objective function, so the algorithm chooses the next values to try in the objective based on maximizing a criterion on the surrogate.

13.2.4 Visualization Integration

In Phase 1, the Grafana visualization dashboard only presents the results of the data-driven anomaly detection core. It does not show the system vulnerability analysis results nor the root cause analysis results. In Phase 2, we added two panels on the dashboard. The first panel automatically pulls results of the vulnerability analysis module and presents them as instructional information at the top of the dashboard. The operators can read these results and decide which node to monitor on the dashboard. The second panel presents the top ranked spider charts of the key contributing factors for price spikes. These charts are generated from the root cause analysis using long-term historical data.

In Phase 1, we used the web-based Large-scale Test Bed (LTB) geographical visualizer to illustrate the LMP changes on the map. In Phase 2, we developed AGVis (Another Grid Visualizer) which is a visualization program that can read simulation data, such as LMP, and output them in a contour heatmap with real geographic locations. AGVis can run independently from LTB as a standalone software. To synchronize with the WISP Grafana visualizer, we shared the simulation data and time stamps between two platforms.

13.3 Conclusions

The software prototype developed in Phase 1 is further modified to accommodate the needs of monitoring a large-scale power system. The team worked on the database restructuring to improve the data fetching efficiency. The multi-thread computing and data buffers were used to improve the computing speed. The parameter tuning process was automated and accelerated using Bayesian optimization and finally the visualization platforms were integrated to present comprehensive detection and analysis results to power grid operators.

Chapter 14

Demonstration

The WISP technology is demonstrated through two large-scale power systems: the Texas synthetic 2000-bus system and the ISO New England (ISO-NE) system. In this chapter, we elaborate for both systems details of the demonstration platforms, the detection performances and the system analysis results.

14.1 Texas 20000-bus System

The Texas synthetic 2000-bus system is built based on public information and statistical analysis of real ERCOT systems [147]. The topology of the Texas synthetic 2000-bus system is shown in Figure 14.1.

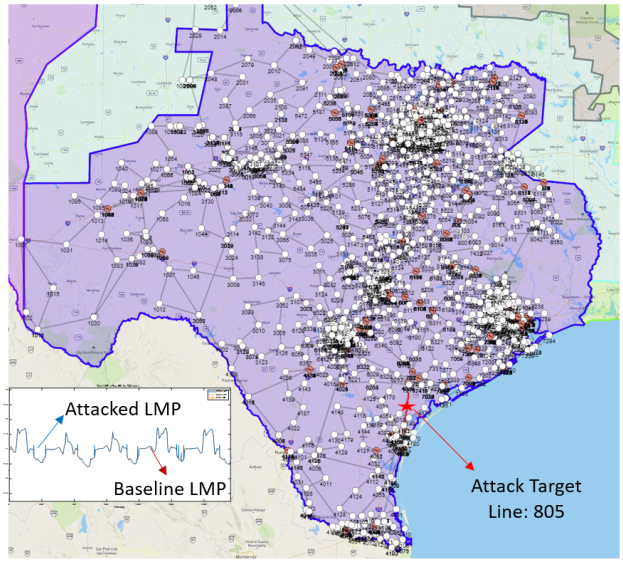


Figure 14.1: The topology of Texas 2000-bus system labeled with cyber-attack targets.

As mentioned in Chapter 12, the FDIAs were implemented on line 805 which connects two heavy load areas. These attacks successfully induced price spikes during the load transition periods.

14.1.1 Data-driven Anomaly Detection Core

The WISP cyber monitoring function is executed through our data-driven anomaly detection core. There are ten individual algorithms implemented in the core: the AutoEncoder, the five pointwise anomaly detectors (the LSTM (long short-term memory), the optimization based probabilistic detection, the gradient boosting regressing (GBR), the joint mean and quantile regression, and the random forests), and the four system anomaly detectors (the isolation forests, the random cut forests, the random forests, and the K nearest neighbor). The scores are fused by the majority voting ensembler. To test the Texas system, we first need to fine tuned the hyper-parameters of the algorithms.

Hyper-parameter Tuning

Most of the algorithms kept their best parameters from the IEEE 39 system tested in Phase 1. The AutoEncoder and the GBR algorithms showed significant improvement after Bayesian optimization based parameter searching.

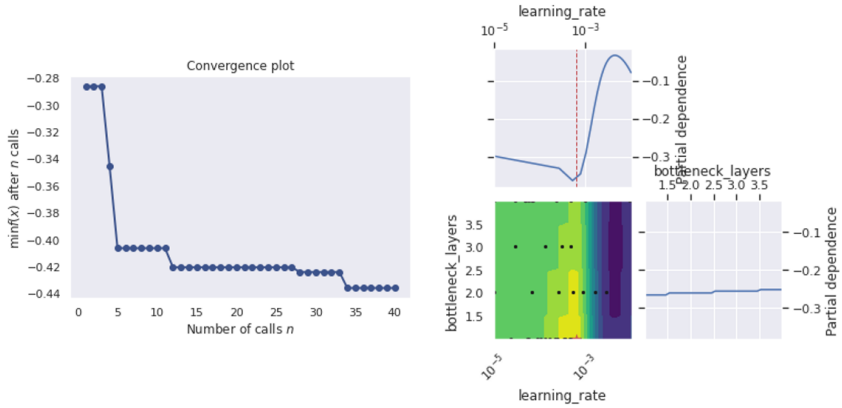


Figure 14.2: The convergence plot of the AutoEncoder detector for the Texas system.

Figure 14.2 shows that the reconstruction error (objective function of the AutoEncoder) was reduced with the number of searching. The learning rate and the number of neurons in the bottleneck layer were finally selected as 0.6e-3 and 1.

Figure 14.3 shows that the prediction error (objective function of the GBR) was reduced with the number of searching. The learning rate and the number of estimators were finally

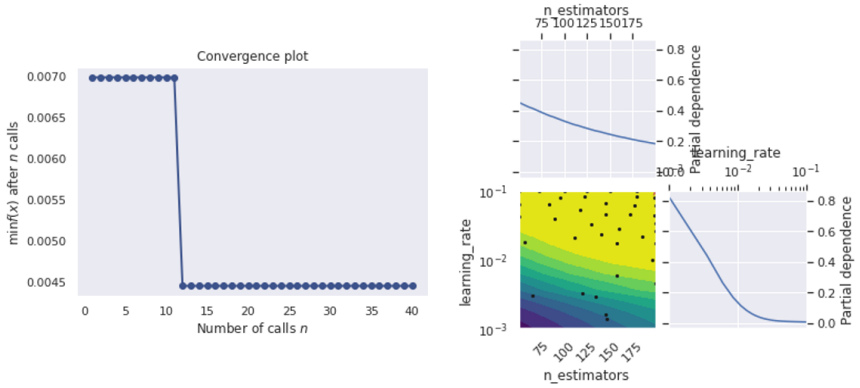


Figure 14.3: The convergence plot of the gradient boosting regression (GBR) detector for the Texas system.

Detection Rate	False Alarm Rate	Computing time
Total attacks: 74	Total false detection: 236	Data fetching time: 33.86s
Total detected attacks: 66	Total benign data: 8566	Data saving time: 0.316s
Detection Rate: 89.2%	False Alarm Rate: 2.76%	Detection Time: 2.275s

Table 14.1: Key performance indexes for the Texas system demonstration.

selected as 0.1 and 200. All the algorithms were trained on 90 days of historical data with no attacks.

Anomaly Detection Performance

The detection core was tested on 30 days of data with FDIAs. Table 14.1 summarized the detection performance for the Texas system and Figure 14.4 illustrated the time series plot fo the detection.

14.1.2 Cyber Vulnerability Analysis

The cyber-vulnerability analysis (CVA) is performed on the Texas system to provide a list of vulnerable components. In the CVA, market data from all sources is assumed to be susceptible to attacks, including line ratings, congestion patterns, generation capacity withholds, market-interface, etc. Namely, all parameters in the ISO's market model are assumed to be attackable. The detailed mathematical model of the CVA can be found in the Phase 1 report.

The CVA identified 5 most vulnerable buses listed as follows: bus 1654, bus 375, bus 1563, bus 1687, and bus 1452. The attack impact of different buses is shown in Figure 14.5.

Figure 14.4: The detection results for major detectors and ground truth labels for 30 days of the Texas system.

The y-axis shows the deviated price in terms of percentage. The x-axis represents different attack degrees, which means how many parameters can be manipulated by attackers. The attack impact increase with a higher attack degree for all buses. It is worth noting that the attack on bus 1654 increased more sharply than others when the attack degree was larger than 7. The above 5 most vulnerable buses were identified based on the average value of the attack impact. Other vulnerability analyses can be done similarly. For example, the 5 most vulnerable lines were line 2778, line 1755, line 3201, line 89, and line 3103. The vulnerable lines were considered where cyber-attacks can achieve more profits by modifying the congestion status. The above vulnerability scan assumed of 15% attack penetration level.

14.1.3 System Visualization

To deliver the cyber monitoring and CVA results to the operators, the WISP software replies on the Grafana time-series visualizer and the AGVis geographical visualizer. The visualization results are shows in Figure 14.6, Figure 14.7 and Figure 14.8.

Figure 14.6 presents the top five panels of the Grafana dashboard. The first panel shows the overall system introduction and the CVA results as additional guidance information for

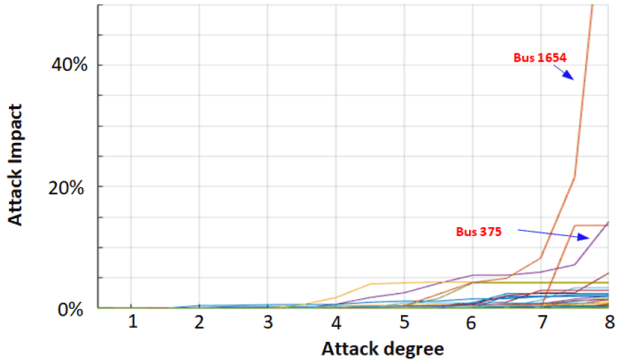


Figure 14.5: CVA results.

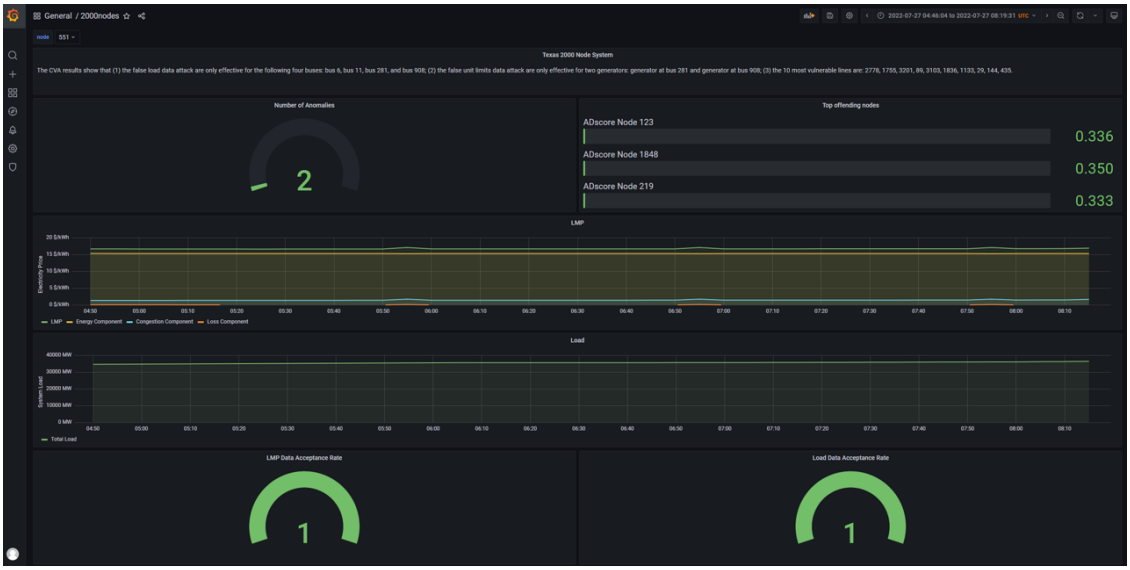


Figure 14.6: The top five panels of the Grafana dashboard for the Texas system.

Figure 14.7: The bottom three panels of the Grafana dashboard for the Texas system.

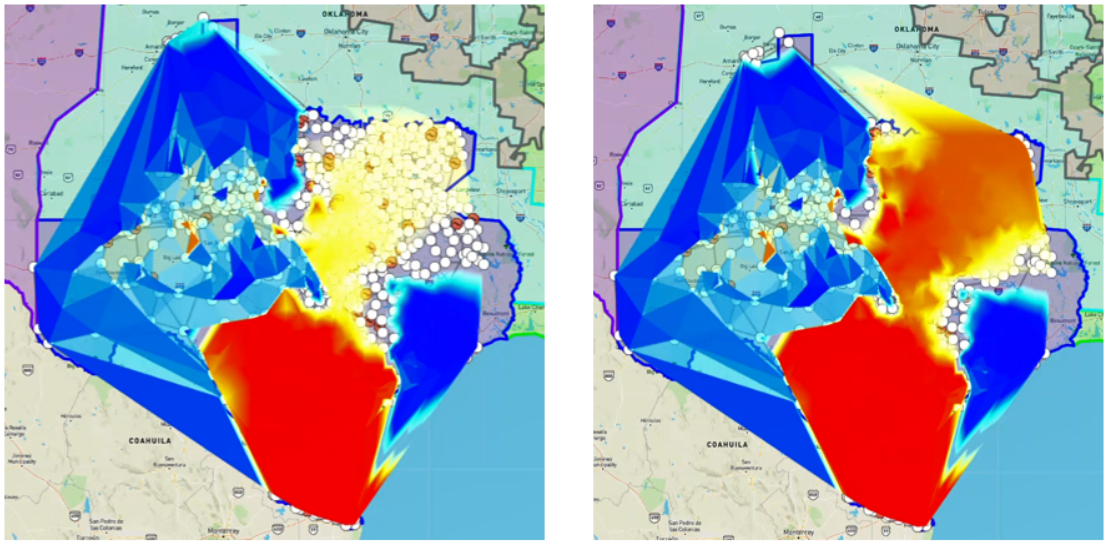


Figure 14.8: The AGVis contour maps before (left) and after (right) attack for the Texas system.

the operators. The second panel contains two sub-panels. The left sub-panel shows the total number of anomalies in the given time window while the right sub-panel shows the three top ranking offensive nodes for potential attack targets. The third panel shows the LMP data (total LMP, energy component, congestion component and loss component) of the selected monitoring node in the given time window. The forth panel shows the total system demand data in the given time window. The fifth panel shows the data acceptance rate for LMP and demand data, computed as the total received data divided over the total expected data. This is to indicate if there are any missing data points in the given time window.

Figure 14.7 presents the bottom three panels of the Grafana dashboard. The first panel shows the binary results (0 means benign and 1 means attack) of the system level anomalies detected from any nodes. The second panel shows the anomaly score breakdown for each individual detector and for the selected monitoring node. The last panel shows the histogram of the raw LMP data and the anomaly scores in the given time window.

Figure 14.8 presents the LMP contour maps before and after cyber attacks. The cooler colors point to lower LMP and the warmer colors point to higher LMP. When line 805 got congested, the LMP prices of the nodes at the top right area increased dramatically.

14.2 ISO-NE System

ISO New England (ISO-NE) is an independent, non-profit Regional Transmission Organization (RTO) serving Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island,

and Vermont. The ISO-NE system consists of 1203 buses and its topology is shown in Figure 14.9. In this section, we performed anomaly detection on randomly selected 100 nodes of the

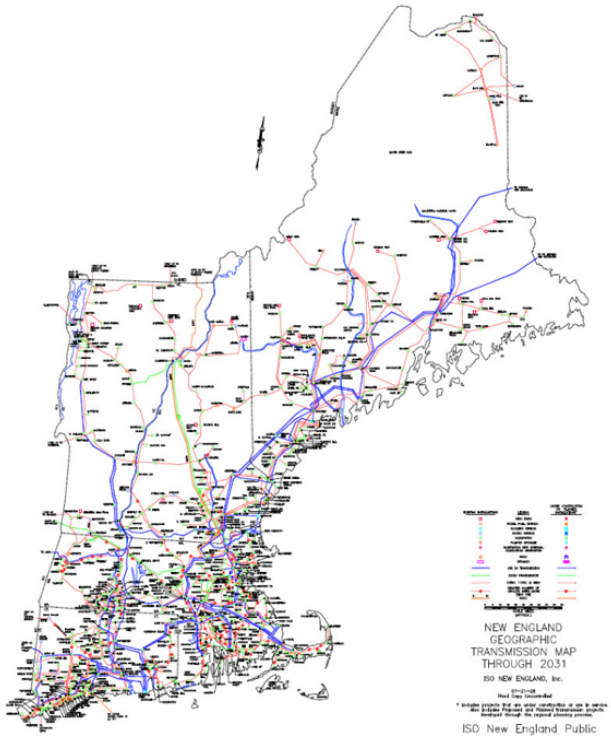


Figure 14.9: The topology of ISO-NE system [5].

ISO-NE system due to the time-consuming process to accumulate historical data for model training. Different from the Texas system, we do not have the detailed system and component models of the ISO-NE system. Thus, we cannot simulate cyber-attacks nor perform the cyber vulnerability analysis. However, there are more available data in the historical database which allows us to perform a thorough root cause analysis for the price spikes. In this section, we provide demonstration results for the data-driven detection core and root cause analysis.

14.2.1 Data-driven Anomaly Detection Core

Similar to the Texas system, we first need to fine tuned the hyper-parameters of the algorithms for the ISO-NE system.

Hyper-parameter Tuning

Most of the algorithms kept their best parameters from the IEEE 39 system tested in Phase 1. The AutoEncoder and the GBR algorithms showed significant improvement after Bayesian optimization based parameter searching.

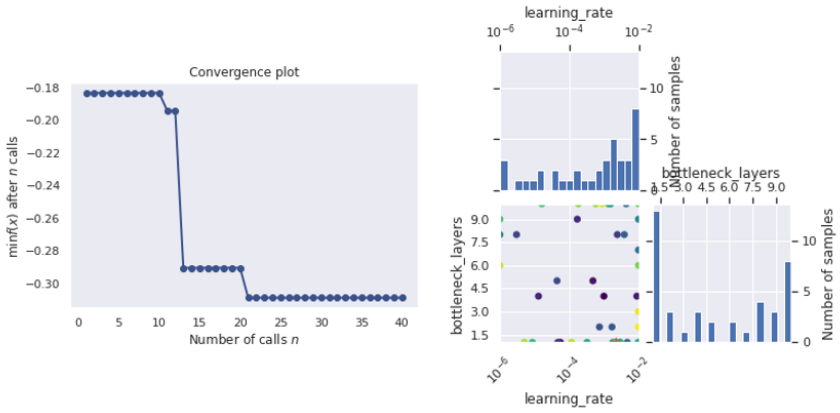


Figure 14.10: The convergence plot of the AutoEncoder detector for the ISO-NE system.

Figure 14.10 shows that the reconstruction error (objective function of the AutoEncoder) was reduced with the number of searching. The learning rate and the number of neurons in the bottleneck layer were finally selected as 2.2e-3 and 1.

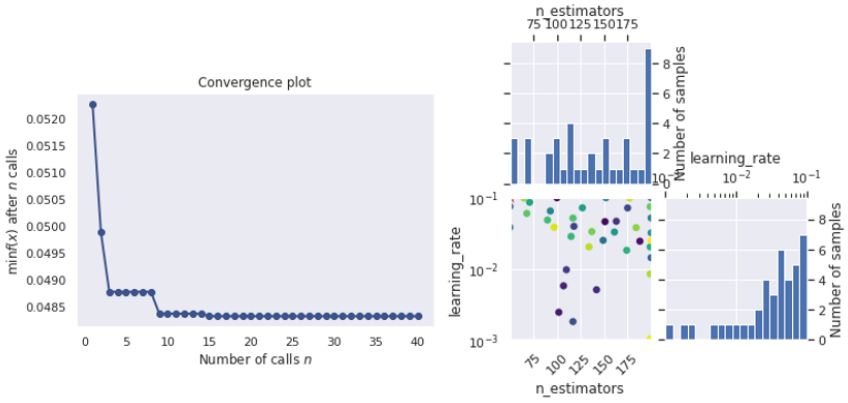


Figure 14.11: The convergence plot of the gradient boosting regression (GBR) detector for the ISO-NE system.

Figure 14.11 shows that the prediction error (objective function of the GBR) was reduced with the number of searching. The learning rate and the number of estimators were finally

Detection Rate	False Alarm Rate	Computing time
Not applicable	Total false detection: 74 Total benign data: 4370 False Alarm Rate: 1.69%	Data fetching time: 8.432s Data saving time: 0.084s Detection Time: 1.894s

Table 14.2: Key performance indexes for the ISO-NE system demonstration.

selected as 0.1 and 50. All the algorithms were trained on 90 days of historical data with no attacks.

Anomaly Detection Performance

The detection core was tested on 15 days of data subsequent to the training data with no attacks. Table 14.2 summarized the detection performance and Figure 14.12 illustrated the time series plots of the detection.

Figure 14.12: The detection results for major detectors and ground truth labels for 15 days of the ISO-NE system.

Note that since there is no attack data, we cannot evaluate the detection accuracy performance for the ISO-NE system. Since most of the false alarms came from the physics-induced

price spikes, we provide, in the following subsection, key contributing factors of the spikes through root cause analysis based upon long-term historical data.

14.2.2 Root Cause Analysis

For nearly 25 years, New England's wholesale electricity markets have attracted billions of dollars in private investment in some of the most efficient, lowest-emitting power resources in the country—providing reliable electricity every second of every day, lowering wholesale prices. In 2021, natural-gas-fired generation, nuclear, other low- or no-emission sources, and imported electricity (mostly hydropower from Eastern Canada) provided of the region's electricity, as seen in Figure 14.13. The total generation by renewable energy are 12.44%, including 4% by wind and 3% by solar.

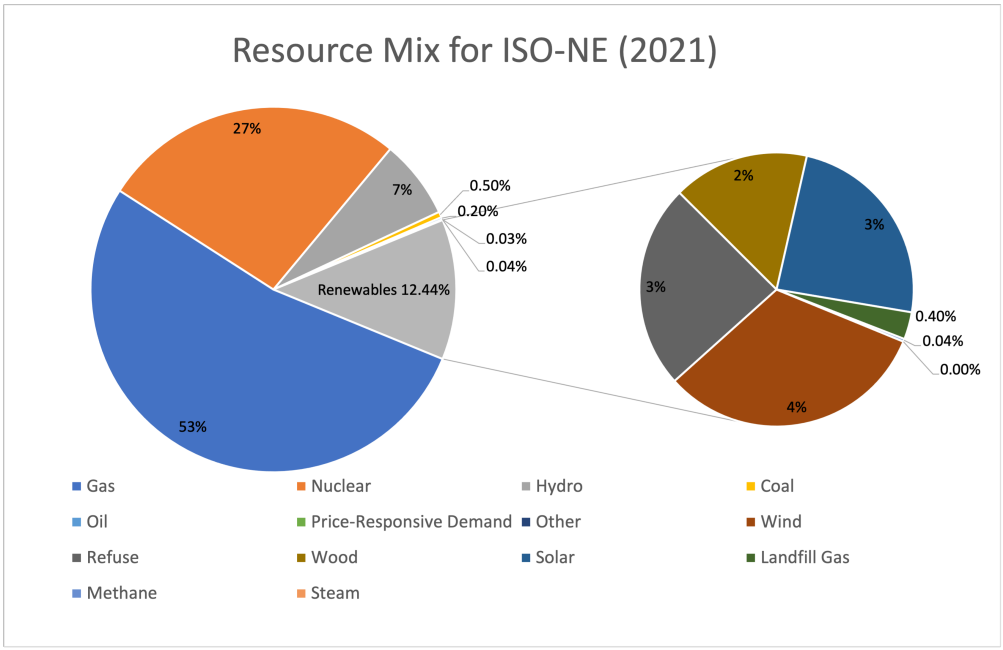


Figure 14.13: Resources Mix for ISO-NE (2021)

Pricing in the ISO-NE wholesale electricity marketplace is calculated at individual generating units, about 900 load nodes, eight load zones (aggregations of load nodes), and the Hub (a collection of locations in central New England where little congestion is evident). Figure 14.14 depicts the eight load zones.

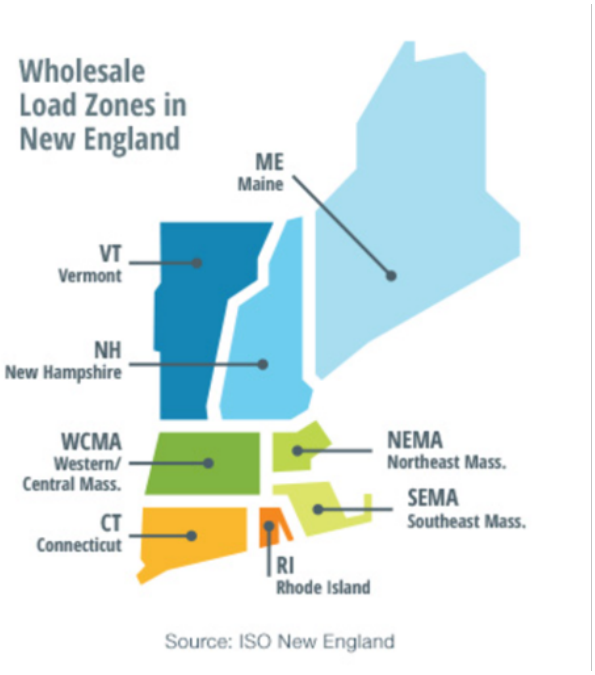


Figure 14.14: Eight Load Zones in ISO-NE (ISO-NE, 2021)

Approach to Root Cause Analysis

We first briefly recap the approach for root cause analysis that has been applied to ISO-NE. The approach follows what has been described in the Phase 1 report, with some modification necessarily for ISO-NE.

Spike detection and data segmentation

Price spikes are defined as prices that exceed certain threshold. The threshold can be a fixed number, or a quantile such as the price at 95 percentiles. Time series price data are segmented to price spike segments and "regular" price segments according to the approach detailed in the Phase 1 report. Figure 14.15 shows the concept of data segmentation and the steps are as below:

- In this step, we first group the events which are close to each other.
- In the next step, we fetch the anomalous data segments. In the anomalous data segments, we select data between $[t_{first} - b_{len}, t_{last} + f_{len}]$. Here, t_{first} and t_{last} depict the first and last occurrence of spike in the grouped event, and b_{len} and f_{len} show the backward and forward size of the data. These segments help us analyze the cause and the effects of spikes.
- In the third step, we divide the rest of the data in normal data segments.

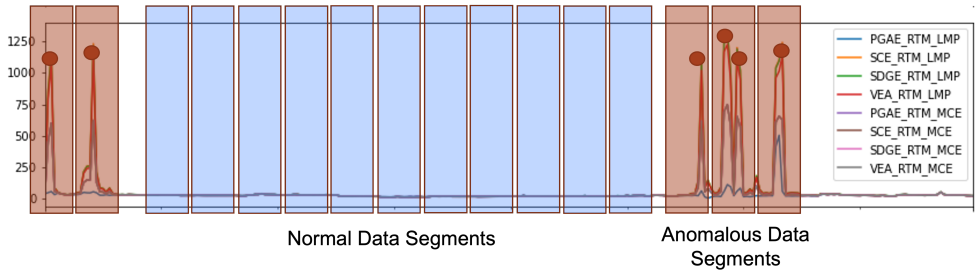


Figure 14.15: Data Segmentation for Price Spikes

Using the above steps, we divided the data into hour-long segments to identify key features related to a price spike event. We kept a buffer of 30 minutes between all the segments to avoid data overlap for accurate modeling. For every segment, maximum MCE was used as the price label and three statistics were calculated for each of the data feature. The mean (μ) estimates the average value of the feature, and the gradient statistics compute the changes in features during that time interval. We used two gradient statistics, g_{avg} and g_{max} that report average and maximum change in the feature value.

Root Cause Identification

We have explored several machine learning techniques to identify and extract the key factors that causes the price spikes in section, including self-organizing map (SOM), autoencoder and random forest. For ISO-NE data, we use autoencoder for identification of features that cause the price spikes. Additionally, we implement clustering analysis on the high dimensional data with different features, to group the time series price and feature data in such a way that data points in the same group are more similar to each other than to those in other groups.

Autoencoder

Autoencoder is an unsupervised deep learning technique used to learn a low-dimensional latent representation of the high-dimensional data. The latent representation is then used to recreate the input feature set. The reconstruction error between the input and the output feature set is a key metric to identify features correlated with price spikes. The process requires training the autoencoder using the non-spike data segments and then passing the spike data segments through the autoencoder. The resulting set of the reconstruction errors are likely to be high for those features which are highly correlated with the price spikes.

Clustering

Besides autoencoder, we also implement clustering analysis on the features. Clustering is the task of grouping a set of data points in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters). Price segments that are driven by the same features in the same way are grouped together. When a new price spike appears, one can easily assign the new spike to a cluster and identify the main root causes for that spike.

ISO-NE Data Description and Price Spike Detection

Data Description

We collected market data from ISO-NE for 2020 and 2021. The LMPs and the three components of LMP, energy component, congestion component, and loss component, are downloaded from ISO-NE for the eight load zones and hub. In this analysis, we only focus on identifying the root causes that impact the marginal cost of energy component. Figure 14.16 shows the energy component of LMP over time in 2020 and 2021, as well as its histogram. Compared with the energy prices in CAISO in the Phase 1 report, energy prices in ISO-NE are lower in general, and the price spikes are sparser. The prices spikes happen most in summer months, and some in winter. Compared with 2020, 2021 has experiences more price spikes, shown in Figure 14.17.

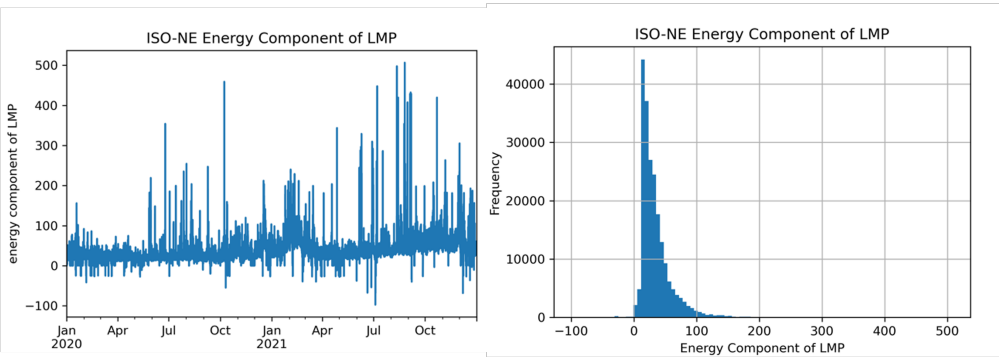


Figure 14.16: Plot and Histogram of Energy Component of LMP in ISO-NE

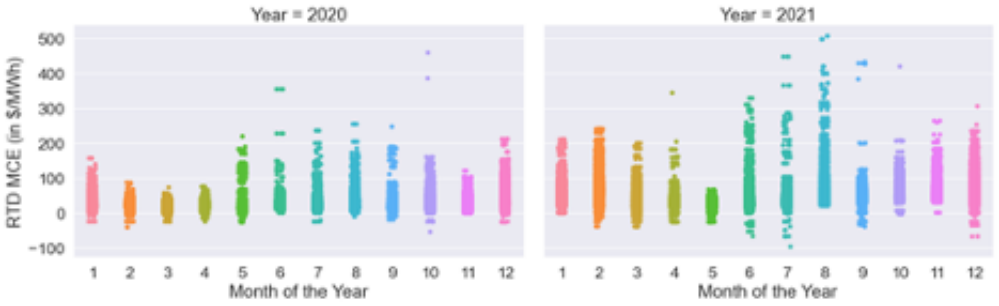


Figure 14.17: ISO-NE Energy Component for ISO-NE by Month and Year

Price Spike Detection

We use 95 percent quantile as the threshold for moderate price spikes, and 99 percent quantile as the threshold for high price spikes. For 2020-2021 data, 223 spikes are detected, including 3 spikes that lasted longer than 5 hours. Figure 14.18 shows an example of price spike segments on Feb 3, 2021, with one spike (in brown) that lasted over 7 hours. Figure 14.19

shows that most of the duration are less than 2 hours. Spikes that happen in winter vary the most in duration, no matter it is in morning, or midday, evening or night. Figure 14.20 shows the number of spikes happen in each hour by season. Winter early morning and evenings and summer midday have more price spikes.

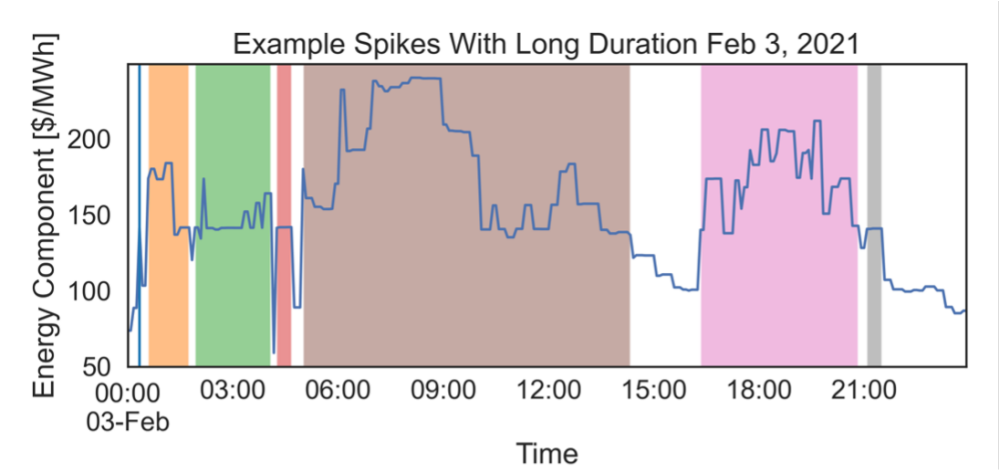


Figure 14.18: Example Price Spike Segments in Feb 03, 2021

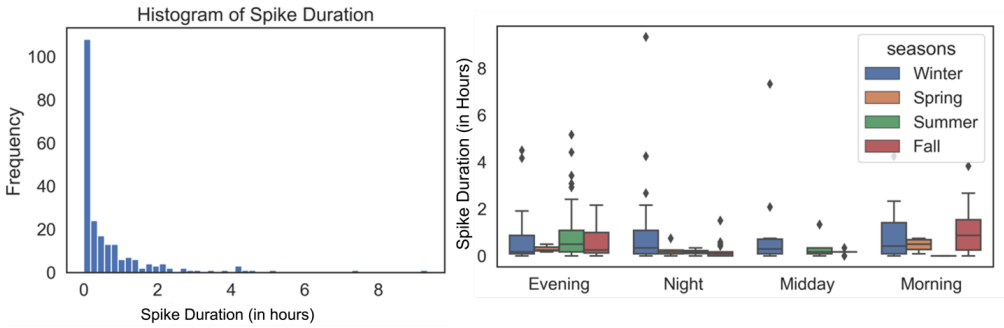


Figure 14.19: Spike Duration: Histogram Plot and Boxplot over Seasons

State Space Representation of Price Segments

State space representation is calculated for every price segment: maximum energy component is used as the price label and several statistics were calculated for each of the data feature, including the mean that estimates the average value of the feature, the standard deviation that estimates the variation of the feature, two gradient statistics, g_{avg} and g_{max} that report the average and maximum change in the feature value. This resulted in data set consisting of over 100 feature vectors. Autoencoder and clustering analysis are applied in these 100+ feature vectors to identify the root causes for price spikes.

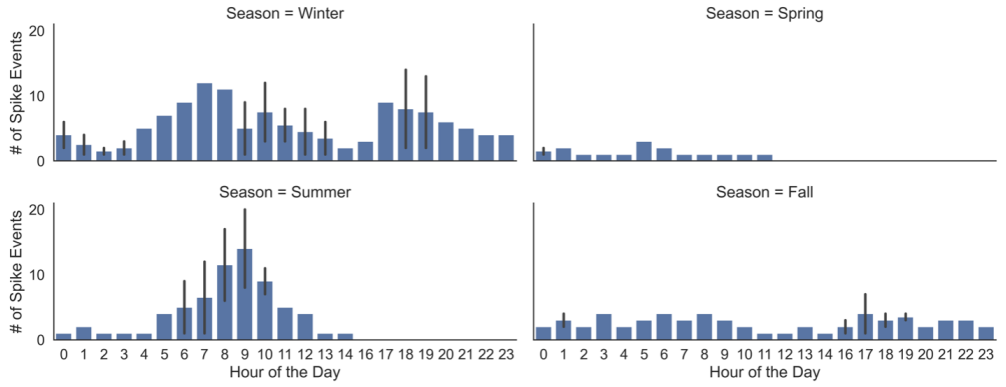


Figure 14.20: Spikes Events by Season and Hour of Day

Feature Identification and Extraction Results

Exploratory Data Analysis Results

We have performed exploratory data analysis on the impact of the features on price spikes. Figure 14.21 to Figure 14.25 are examples showing how these features impact the price spikes. In summary:

- Demands for spike periods are significantly higher than those of non-spike periods.
- Demand forecast errors are not significantly relevant to the price spikes.
- Wind power (including real time wind as well as wind forecasts) does not show significant relevance with the price spikes from preliminary analysis.
- Solar, hydro-power, and natural gas generation have significant impact on price spikes.
- Reserve prices have significant impact on price spikes.
- Whether the export or import limits have been hit has significant impact on the price spikes.
- Marginal fuel type influences the prices, but how it causes the price spikes needs further investigation.

Autoencoder Results

We implement autoencoder to identify the key features that cause the price spikes. We first implement autoencoder on the entire data set, then on data for each season to further investigate how each feature impact the prices in different seasons. Figure 14.26 shows the reconstruction errors for the top 40 factors when implementing autoencoder on the entire data set. The top features are the reserve prices, whether the import/export limits have

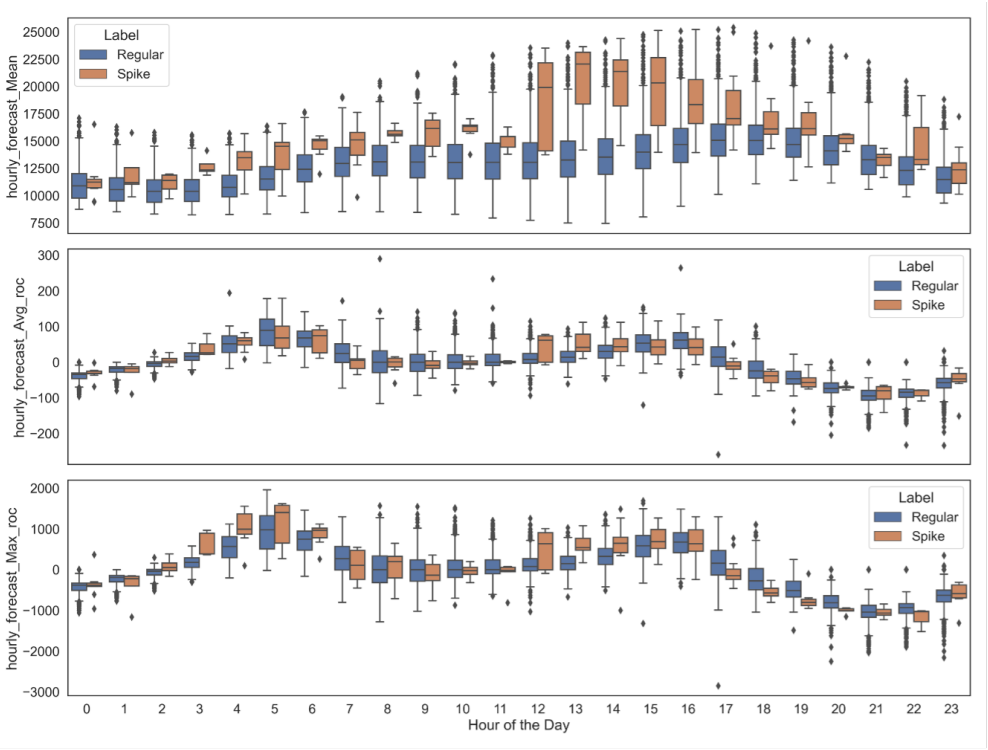


Figure 14.21: Hour-ahead Forecast Error for Price Spikes and Non-spikes

been hit, the congestion component, generation of hydropower, wind power forecast errors, day ahead demand forecast errors, the ratio of renewable generation etc. Figure 14.27 and Figure 14.28 shows the comparison of construction errors for training and test data sets, and the price spikes and non-spikes, respectively. We can see that the reconstruction errors for non-spikes are significantly different for the top features.

To investigate how each feature impact the prices differently by season, we also performed autoencoder for each season. Figure 14.29 to Figure 14.32 shows the reconstruction errors for the top features by season. We can see that the reserve price, export/import limit hits, demand and demand forecast errors remain the top features across different seasons. There are some features that work specifically for one season. In summer, the solar generation level and variation contributes as the third top factor to the price spikes. In fall, nuclear generation is one of the top factors, and in winter, generation of hydropower and wind play significant impact on the price spikes.

Clustering results

Clustering analysis was done to the entire datasets to group price segments with similar features. Price spikes in different clusters will be caused by different key features. We first use elbow method to determine the optimal number of clusters, and then performed k-means clustering to cluster the entire data set. Figure 14.33 shows that the number of clusters is

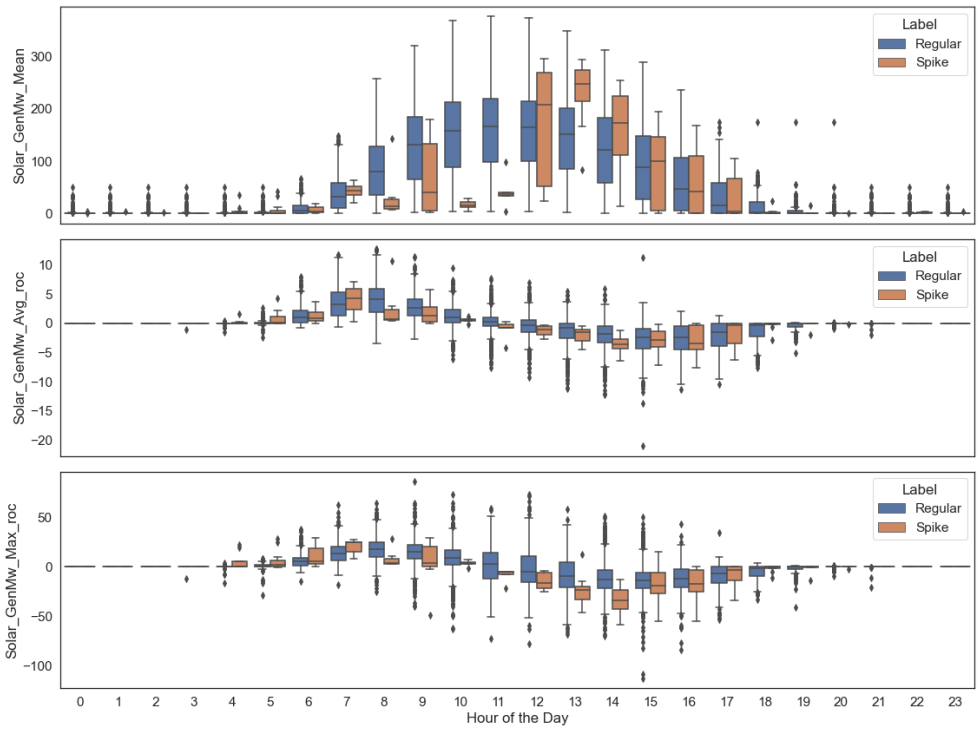


Figure 14.22: Solar Generation for Price Spikes and Non-spikes

optimal approximately at 8 clusters. We cluster the data sets to 8 clusters. Figure 14.34 shows the top 6 clusters and the top 10 features for each cluster. We can see that in Cluster 5 (in green), reserve prices, solar generation, wind generation and import hits play more important roles than in other clusters. For cluster 0 (in blue), day ahead demand forecast errors, wind Generation and variation in congestion component have larger impact.

Summary

As the electricity grid continues to evolve with increases in renewable penetration, electricity markets will continue to see changes in price behaviors – price spikes, volatility, and negative prices. We proposed a machine learning-based approach that provides a fast and robust methodology to automatically identify the primary drivers for price spikes using publicly available data only. The Phase 1 report has applied this approach to CAISO and in this section, we apply it to ISO-NE. The raw data set downloaded from ISO-NE were used to identify root causes behind price spikes. It consists of load, renewable forecasts and their forecast errors, reserve prices, and system conditions such as the import and export limit hits, etc. Machine learning algorithms, auto-encoders and clustering were used to identify the data features that have significant impact on the market outcomes, resulting in price spikes.

These analysis helped concluded that the price spikes are highly correlated with reserve

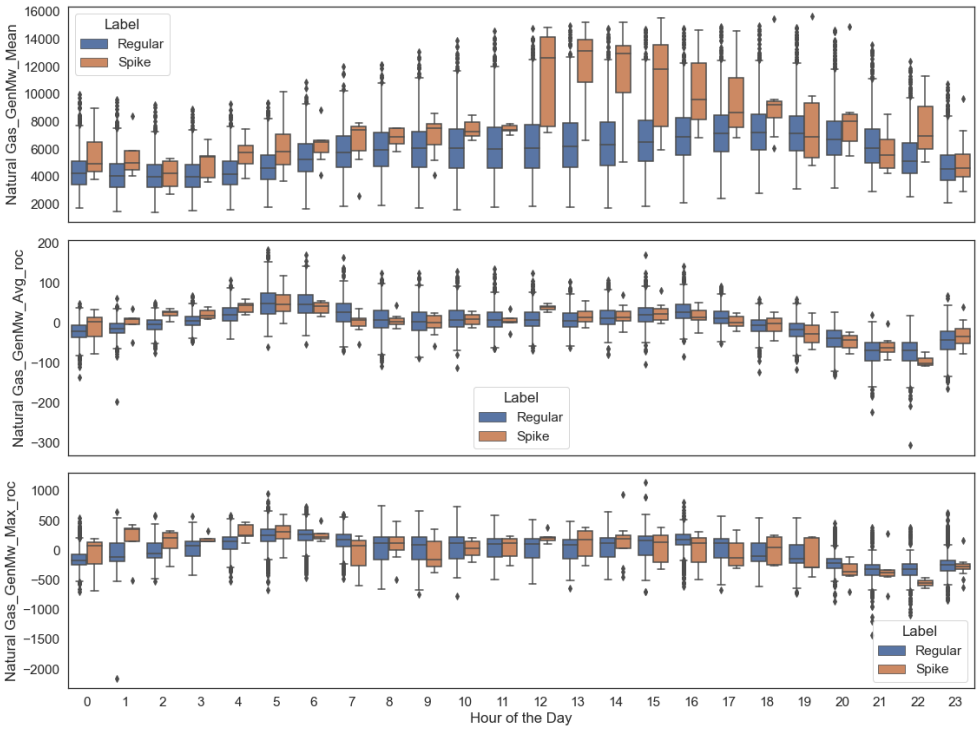


Figure 14.23: Natural Gas Generation and Price Spikes and non Price Spikes

prices. The mean, standard deviation and the average rate of change, as well as the max rate of change of reserve prices all have significant impact on the price spikes. Whether the power flow exceeds the import and export limits also plays an important role in price spikes. Load and load forecasts, renewable ratios, wind and solar generations are also the key features that cause the prices spikes. Some features behave differently in different seasons and have different impact on the electricity prices. For example, solar generation and its variation is a top factor that causes the price spikes in summer. The price spikes are resulted from complex interactions between various data features, each with their own significance that evolves over different hours and seasons. Our clustering analysis grouped the data sets into clusters containing data points with similar price drivers. This allows us to quickly assign one new price spike segment to a cluster and identify the key causes for that price spike.

14.2.3 System Visualization

To deliver the cyber monitoring and root cause analysis results to the operators, the WISP software replies on the Grafana time-series visualizer. The visualization results are shows in Figure 14.35 and Figure 14.36.

Figure 14.35 presents the top five panels of the Grafana dashboard. The first panel shows

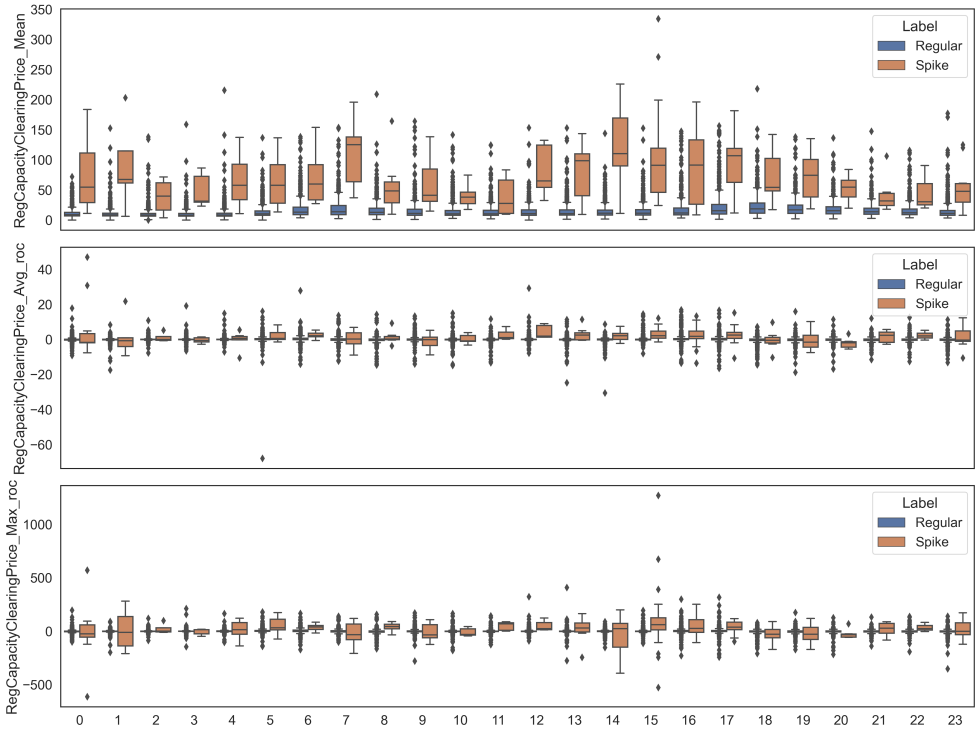


Figure 14.24: Regulation Capacity Clearing Prices and Price Spikes and non Spikes

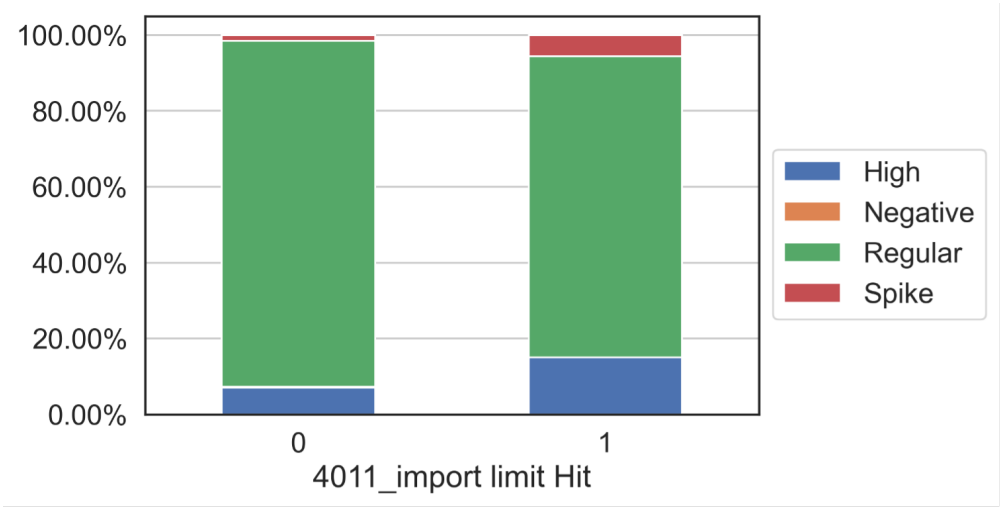


Figure 14.25: Power Import Exceeding Limit and Price Spikes

the overall system introduction as additional information for the operators. The second panel contains two sub-panels. The left sub-panel shows the total number of anomalies

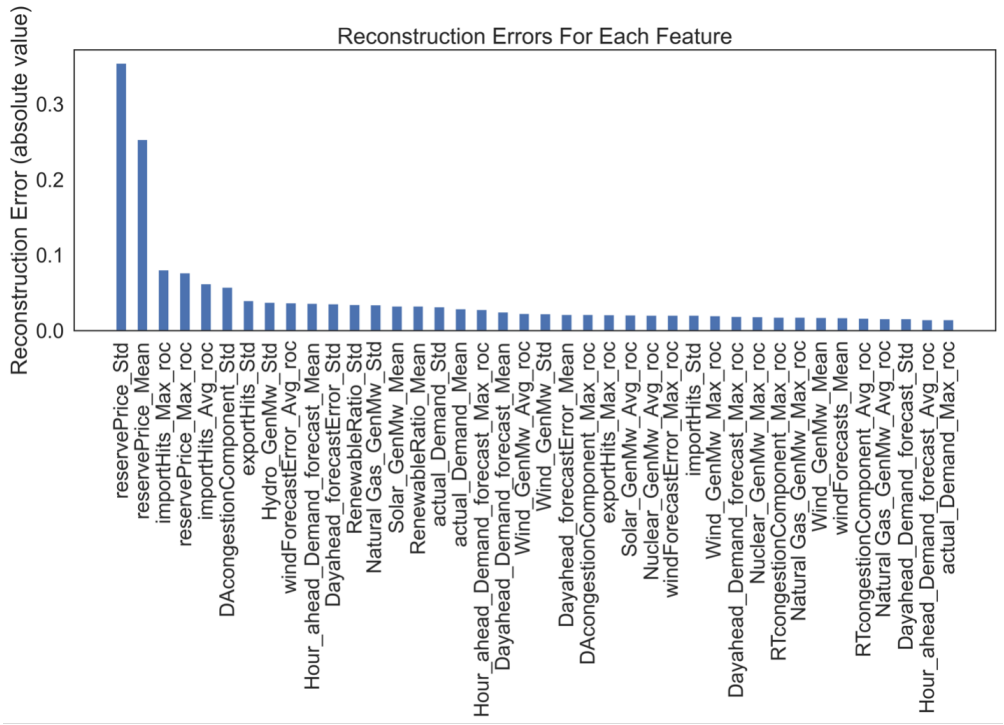


Figure 14.26: Reconstruction Errors (absolute value) for the Top 40 Features for the Entire Datase

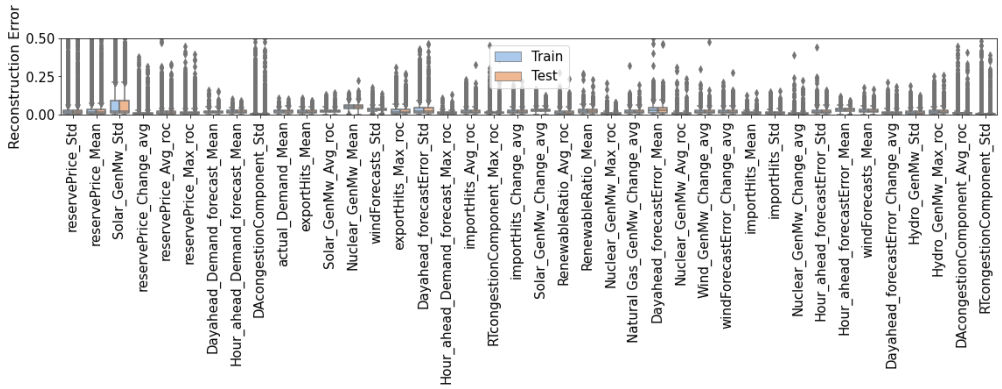


Figure 14.27: Reconstruction Error Comparison for Training and Testing Dataset

in the given time window while the right sub-panel shows the three top ranking offensive nodes for potential attack targets. The third panel shows the LMP data (total LMP, energy component, congestion component and loss component) of the selected node in the given time window. The forth panel shows the total system demand data in the given time window. The fifth panel shows the data acceptance rate for LMP and demand data, computed as the

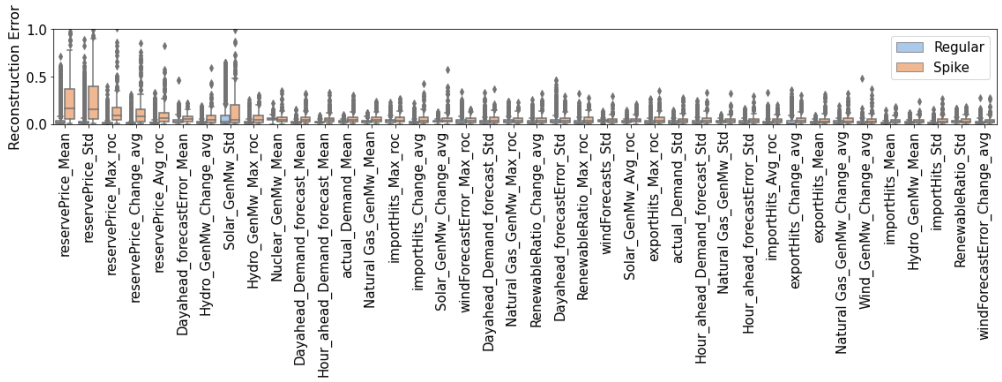


Figure 14.28: Reconstruction Error Comparison for Spike and non-Spike Price Segments

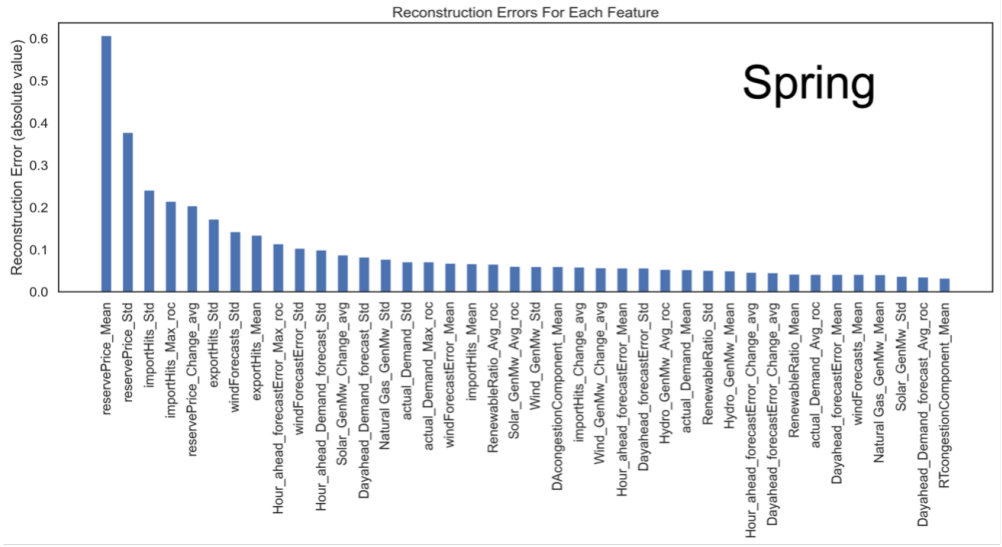


Figure 14.29: Reconstruction Errors for Top Features in Spring

total received data divided over the total expected data. This is to indicate if there are any missing data points in the given time window.

Figure 14.36 presents the bottom four panels of the Grafana dashboard. The first panel shows the binary results (0 means benign and 1 means attack) of the system level anomalies detected from any nodes. The second panel shows the anomaly score breakdown for each individual detector and for the selected monitoring node. The third panel shows the histogram of the raw LMP data and the anomaly scores in the given time window. The last panel shows the spider charts for the top six clusters of price spikes. This is to help the operators understand potential physical and operational causes for the observed price spikes.

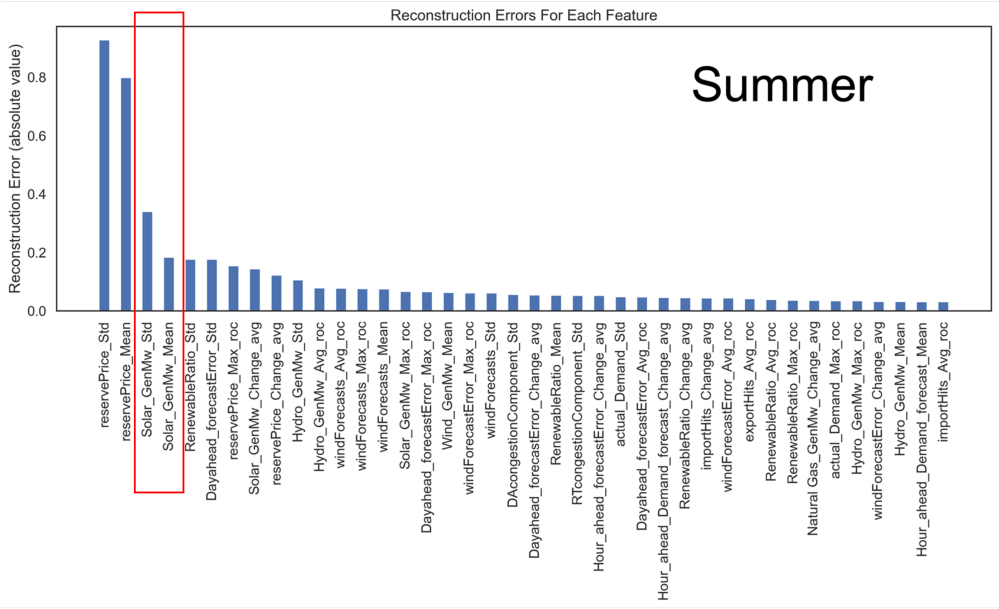


Figure 14.30: Reconstruction Errors for Top Features in Summer

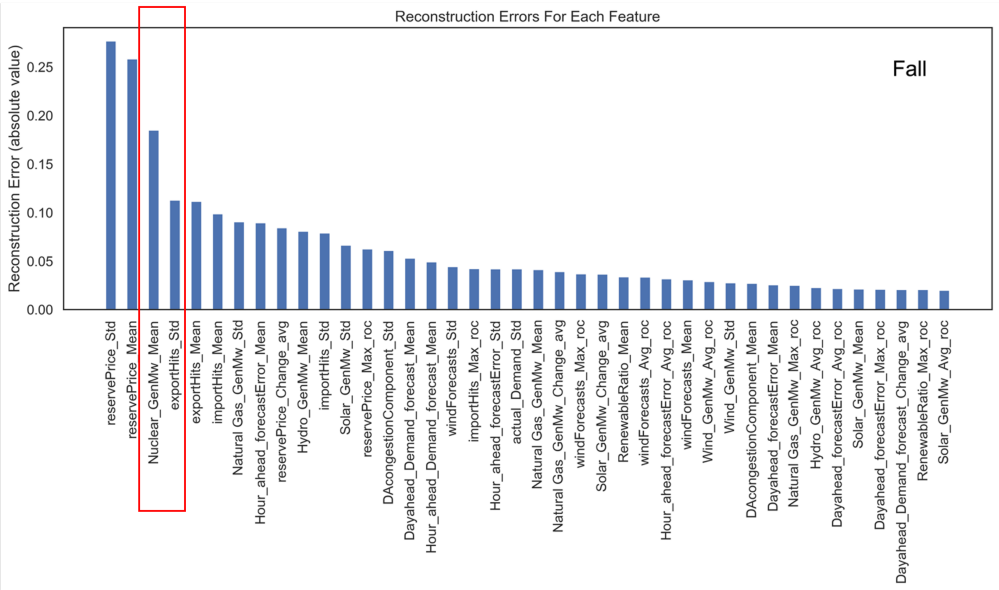


Figure 14.31: Reconstruction Errors for Top Features in Fall

14.3 Conclusions

The WISP technology was demonstrated on the Texas 2000-bus system and the ISO-NE system. Overall, the detection rate is above 89% and the false alarm rate is below 3%.

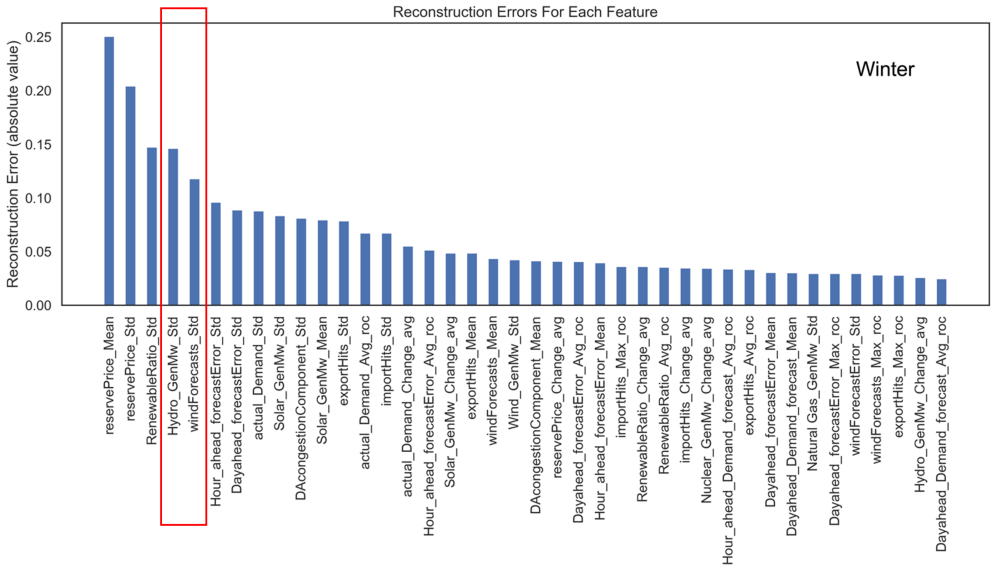


Figure 14.32: Reconstruction Errors for Top Features in Winter

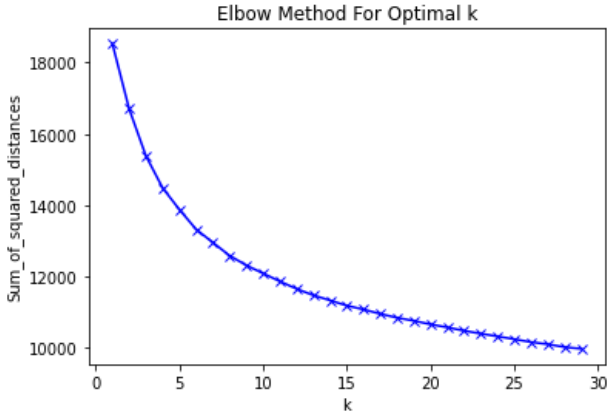


Figure 14.33: Elbow Method to Determine the Optimal Number of Clusters

The total end-to-end computing delay is below 37 seconds. It is worth mentioning that the detection rate depends on the impacts of the cyber-attacks. If we focus only on the attacks that cause major disturbances in the market, we can achieve very high detection rate. For our demonstration, we did not explicitly filter out the attacks with low impacts. Thus the detection rate and false alarm rate are satisfactory but not perfect. In real implementations, we provide the threshold selection function to allow the operators to decide the sensitivity of the detectors. The more sensitive detector can detect low-impact attacks but with higher false alarms, vice versa. For real-time implementation, the data downloading speed is largely constrained by the remote utility database server. Based on our test for the ISO-NE system,

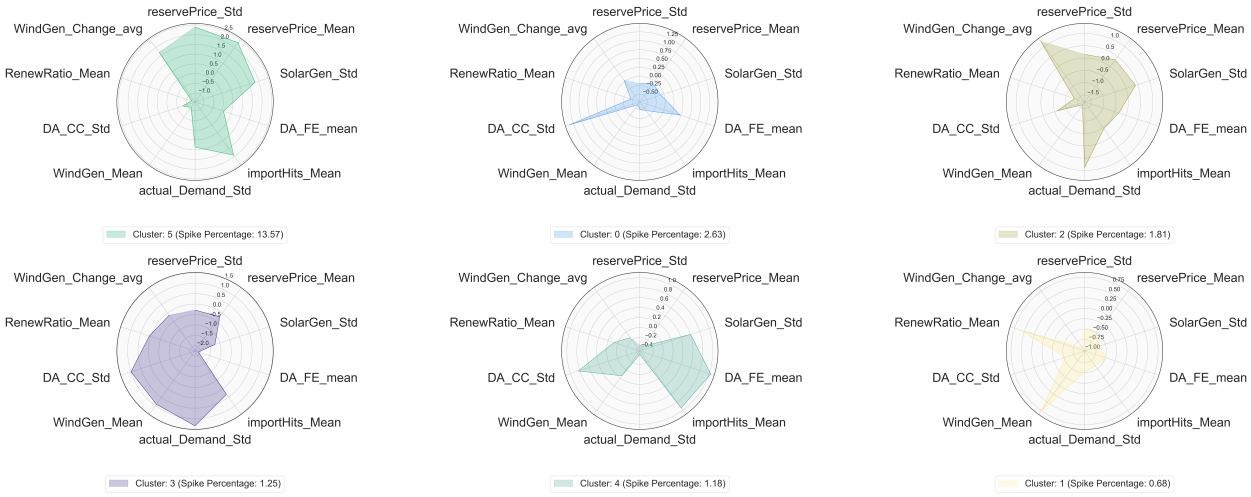


Figure 14.34: Top 6 Clusters

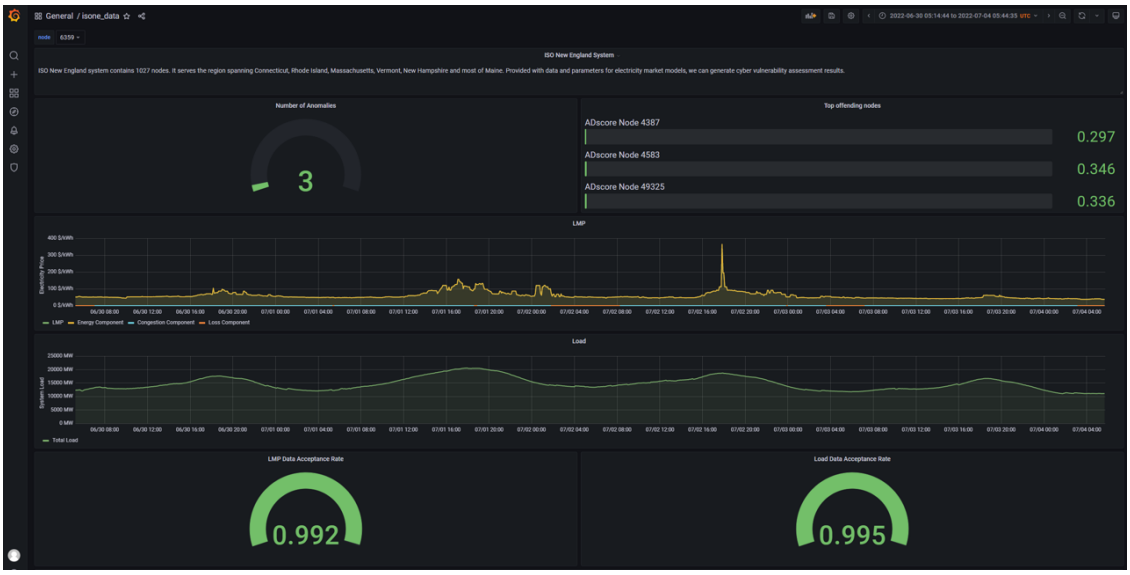


Figure 14.35: The top five panels of the Grafana dashboard for the ISO-NE system.

the detection results can still be ready within the five minutes time interval.

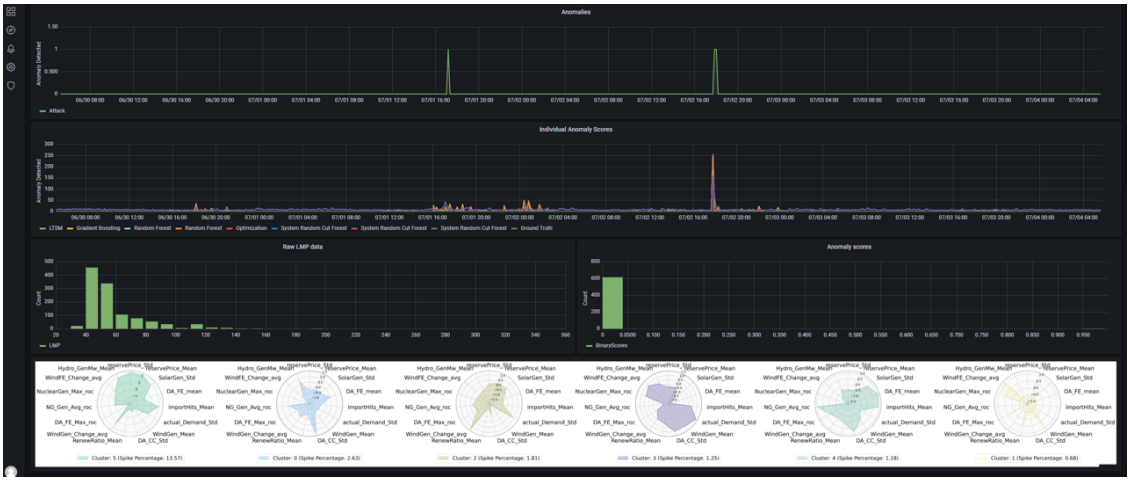


Figure 14.36: The bottom four panels of the Grafana dashboard for the ISO-NE system.

Chapter 15

Conclusions - Phase II

The objective of WISP Phase 2 is to test and demonstrate the WISP software on realistic real-world scale power systems. We first performed red team testing for the WISP software platform to identify cyber vulnerabilities and enforce the cyber hardening solutions. We then selected the Texas synthetic 2000-bus system and the ISO-NE system as the two demonstration use cases. For the Texas system, we implemented the system models and parameters on the electricity market simulator. To generate cyber-attack data, we improved the simulator optimizer and modified the false data injection attacks to apply to the DC power flow. We searched and selected the most impactful attack locations and attack time periods for the final demonstration. For the WISP software, we improved the database structure, implemented parallel computing and automated parameter tuning to achieve the most efficient and accurate anomaly detection and enable applications to different systems and performance requirements. The visualization platforms were integrated to present the detection and analysis results simultaneously to better facilitate the operators in decision making. The demonstration performance met the requirements for real-time cyber monitoring of large-scale systems. In summary, we have successfully demonstrated the effectiveness of the WISP software in providing timely anomaly detection and diagnostic results to the operators.

Bibliography

[1] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, pp. 413–422, IEEE, 2008.

[2] I. N. England, "Iso new england maps and diagrams - load zones."

[3] A. Géron, "Hands-on machine learning with scikit-learn and tensorflow: Concepts," *Tools, and Techniques to build intelligent systems*, 2017.

[4] CAISO, "Welcome to the California ISO." `http://www.caiso.com/Documents/WelcomeToTheISO-ParticipantSlides_ToolKit.pdf`, 2020.

[5] I. N. England, "Iso new england maps and diagrams - geographical transmission system."

[6] N. Sayegh, I. H. Elhajj, A. Kayssi, and A. Chehab, "Scada intrusion detection system based on temporal behavior of frequent patterns," in *MELECON 2014-2014 17th IEEE Mediterranean Electrotechnical Conference*, pp. 432–438, IEEE, 2014.

[7] A. Carcano, A. Coletta, M. Guglielmi, M. Masera, I. N. Fovino, and A. Trombetta, "A multidimensional critical state analysis for detecting intrusions in scada systems," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 2, pp. 179–186, 2011.

[8] S. Cheung, B. Dutertre, M. Fong, U. Lindqvist, K. Skinner, and A. Valdes, "Using model-based intrusion detection for scada networks," in *Proceedings of the SCADA security scientific symposium*, vol. 46, pp. 1–12, Citeseer, 2007.

[9] U. K. Premaratne, J. Samarabandu, T. S. Sidhu, R. Beresh, and J.-C. Tan, "An intrusion detection system for iec61850 automated substations," *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2376–2383, 2010.

[10] Y. Yang, K. McLaughlin, S. Sezer, T. Littler, E. G. Im, B. Pranggono, and H. Wang, "Multiattribute scada-specific intrusion detection system for power networks," *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1092–1102, 2014.

[11] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.

[12] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, "A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids," *IEEE Access*, vol. 5, pp. 26022–26033, 2017.

[13] C. Glenn, D. Sterbentz, and A. Wright, "Cyber threat and vulnerability analysis of the us electric sector," tech. rep., Idaho National Lab.(INL), Idaho Falls, ID (United States), 2016.

[14] ICF, "Electric grid security and resilience-establishing a baseline for adversarial threats," 2016. https://www.energy.gov/sites/prod/files/2017/01/f34/Electric%20Grid%20Security%20and%20Resilience--Establishing%20a%20Baseline%20for%20Adversarial%20Threats.pdf.

[15] R. J. Campbell, "Cybersecurity issues for the bulk power system," 2015.

[16] I. Pena, M. Ingram, and M. Martin, "States of cybersecurity: Electricity distribution system discussions," tech. rep., National Renewable Energy Lab.(NREL), Golden, CO (United States), 2017.

[17] C. Greer, D. A. Wollman, D. E. Prochaska, P. A. Boynton, J. A. Mazer, C. T. Nguyen, G. J. FitzPatrick, T. L. Nelson, G. H. Koepke, A. R. Hefner Jr, *et al.*, "Nist framework and roadmap for smart grid interoperability standards, release 3.0," tech. rep., 2014.

[18] H. T. Haider, O. H. See, and W. Elmenreich, "A review of residential demand response of smart grid," *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 166–178, 2016.

[19] M. J. Assante and R. M. Lee, "The industrial control system cyber kill chain," *SANS Institute InfoSec Reading Room*, vol. 1, 2015.

[20] EPRI, "Electric sector failure scenarios common vulnerabilities and mitigations mapping," 2015. https://smartgrid.epri.com/doc/NESCOR%20Common%20Vulnerabilities%20and%20Mitigations%20Mapping%2012-11-15.pdf.

[21] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2011.

[22] C.-C. Sun, C.-C. Liu, and J. Xie, "Cyber-physical system security of a power grid: State-of-the-art," *Electronics*, vol. 5, no. 3, p. 40, 2016.

[23] C.-W. Ten, C.-C. Liu, and G. Manimaran, "Vulnerability assessment of cybersecurity for scada systems," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1836–1846, 2008.

[24] V. Sridharan, *Cyber security in power systems*. PhD thesis, Georgia Institute of Technology, 2012.

[25] I. Darwish, O. Igbe, O. Celebi, T. Saadawi, and J. Soryal, "Smart grid dnp3 vulnerability analysis and experimentation," in *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing*, pp. 141–147, IEEE, 2015.

[26] Z. El Mrabet, N. Kaabouch, H. El Ghazi, and H. El Ghazi, "Cyber-security in smart grid: Survey and challenges," *Computers & Electrical Engineering*, vol. 67, pp. 469–482, 2018.

[27] DHS, "Industrial control systems cyber emergency response team reports," `https://www.us-cert.gov/ics/Other-Reports`.

[28] V. Kekatos, G. B. Giannakis, and R. Baldick, "Grid topology identification using electricity prices," in *2014 IEEE PES General Meeting| Conference & Exposition*, pp. 1–5, IEEE, 2014.

[29] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[30] Y. Yuan, Z. Li, and K. Ren, "Modeling load redistribution attacks in power systems," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 382–390, 2011.

[31] L. Xie, Y. Mo, and B. Sinopoli, "False data injection attacks in electricity markets," in *2010 First IEEE International Conference on Smart Grid Communications*, pp. 226–231, IEEE, 2010.

[32] L. Xie, Y. Mo, and B. Sinopoli, "Integrity data attacks in power market operations," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 659–666, 2011.

[33] L. Jia, R. J. Thomas, and L. Tong, "Malicious data attack on real-time electricity market," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5952–5955, IEEE, 2011.

[34] S. Bi and Y. J. Zhang, "False-data injection attack to control real-time price in electricity market," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 772–777, IEEE, 2013.

[35] M. Roozbehani, M. A. Dahleh, and S. K. Mitter, "Volatility of power grids under real-time pricing," *IEEE Transactions on Power Systems*, vol. 27, pp. 1926–1940, Nov 2012.

[36] J. Giraldo, A. Cárdenas, and N. Quijano, "Integrity attacks on real-time pricing in smart grids: Impact and countermeasures," *IEEE Transactions on Smart Grid*, vol. 8, pp. 2249–2257, Sept 2017.

[37] R. Tan, V. B. Krishna, D. K. Y. Yau, and Z. Kalbarczyk, "Integrity attacks on real-time pricing in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 18, pp. 5:1–5:33, July 2015.

[38] H. Li and Z. Han, "Manipulating the electricity power market via jamming the price signaling in smart grid," in *2011 IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 1168–1172, Dec 2011.

[39] X. Yang, X. Zhang, J. Lin, W. Yu, X. Fu, and W. Zhao, "Data integrity attacks against the distributed real-time pricing in the smart grid," in *2016 IEEE 35th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, Dec 2016.

[40] C. Barreto and A. A. Cárdenas, "Detecting fraud in demand response programs," in *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 5209–5214, Dec 2015.

[41] C. Barreto, A. A. Cárdenas, N. Quijano, and E. Mojica-Nava, "Cps: Market analysis of attacks against demand response in the smart grid," in *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, (New York, NY, USA), pp. 136–145, ACM, 2014.

[42] Y. Liu, S. Hu, and T. Ho, "Vulnerability assessment and defense technology for smart home cybersecurity considering pricing cyberattacks," in *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 183–190, Nov 2014.

[43] T. Wei, B. Zheng, Q. Zhu, and S. Hu, "Security analysis of proactive participation of smart buildings in smart grid," in *2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 465–472, Nov 2015.

[44] A. Mohsenian-Rad and A. Leon-Garcia, "Distributed internet-based load altering attacks against smart power grids," *IEEE Transactions on Smart Grid*, vol. 2, pp. 667–674, Dec 2011.

[45] S. Amini, F. Pasqualetti, and H. Mohsenian-Rad, "Dynamic load altering attacks against power system stability: Attack models and protection schemes," *IEEE Transactions on Smart Grid*, vol. 9, pp. 2862–2872, July 2018.

[46] T. Ryutov, A. Almajali, and C. Neuman, "Modeling security policies for mitigating the risk of load altering attacks on smart grid systems," in *2015 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pp. 1–6, April 2015.

[47] K. Khanna, B. K. Panigrahi, and A. Joshi, "Bid modification attack in smart grid for monetary benefits," in *2016 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, pp. 224–229, IEEE, 2016.

[48] R. D. Tabors and J. B. Cardell, "Ex ante and ex post designs for electric market mitigation: Past and present experience and lessons from california," in *36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the*, pp. 10–pp, IEEE, 2003.

[49] Open Electrical, "Power system analysis software." `https://wiki.openelectrical.org/index.php?title=Power_Systems_Analysis_Software`, Last accessed on 2020-04-29.

[50] L. Tesfatsion, "Open source software (oss) for electricity market research, teaching, and training." `http://www2.econ.iastate.edu/tesfatsi/ElectricOSS.htm`, Last accessed on 2020-04-29.

[51] L. Tesfatsion, "The ames wholesale power market test bed." `http://www2.econ.iastate.edu/tesfatsi/AMESMarketHome.htm`, Last accessed on 2020-04-29.

[52] Z. Vale, T. Pinto, I. Praca, and H. Morais, "Mascem: electricity markets simulation with strategic agents," *IEEE Intelligent Systems*, vol. 26, no. 2, pp. 9–17, 2011.

[53] R. D. Zimmerman and C. E. Murillo-Sanchez, "Matpower (version 7.0)," *[Software] Available at: https://matpower.org*, 2019.

[54] F. Milano, "Power system analysis toolbox." `http://faraday1.ucd.ie/psat.html`, Last accessed on 2020-04-29.

[55] A. L. Ott, "Experience with pjm market operation, system design, and implementation," *IEEE Transactions on Power Systems*, vol. 18, no. 2, pp. 528–534, 2003.

[56] 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, 2012.

[57] W. W. Kotiuga and M. Vidyasagar, "Bad data rejection properties of weughted least absolute value techniques applied to static state estimation," *IEEE Transactions on Power Apparatus and Systems*, no. 4, pp. 844–853, 1982.

[58] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.

[59] B. Stott, J. Jardim, and O. Alsaç, "Dc power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[60] F. Li and R. Bo, "Dcopf-based lmp simulation: algorithm, comparison with acopf, and sensitivity," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1475–1485, 2007.

[61] B. Eldridge, R. P. O'Neill, and A. Castillo, "Marginal loss calculations for the dcopf," *Federal Energy Regulatory Commission, Tech. Rep*, 2017.

[62] T. Zheng and E. Litvinov, "Ex post pricing in the co-optimized energy and reserve market," *IEEE Transactions on Power Systems*, vol. 21, no. 4, pp. 1528–1538, 2006.

[63] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: Mathematical models and approaches," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 570–582, 2015.

[64] R. Tan, V. B. Krishna, D. K. Yau, and Z. Kalbarczyk, "Integrity attacks on real-time pricing in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 2, pp. 1–33, 2015.

[65] R. Deng, G. Xiao, R. Lu, H. Liang, and A. V. Vasilakos, "False data injection on state estimation in power systems—attacks, impacts, and defense: A survey," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 411–423, 2016.

[66] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Distributed internet-based load altering attacks against smart power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 667–674, 2011.

[67] S. Soltan, P. Mittal, and V. Poor, "Protecting the grid against mad attacks," *IEEE Transactions on Network Science and Engineering*, 2019.

[68] L. Jia, J. Kim, R. J. Thomas, and L. Tong, "Impact of data quality on real-time locational marginal price," *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 627–636, 2013.

[69] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad data injection attack and defense in electricity market using game theory study," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 160–169, 2013.

[70] PJM, "Pjm data miner 2." `https://dataminer2.pjm.com/feed/very_short_load_frcst/definition`, Last accessed on 2020-04-29.

[71] X. Wang, T. Zhao, H. Liu, and R. He, "Power consumption predicting and anomaly detection based on long short-term memory neural network," in *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pp. 487–491, IEEE, 2019.

[72] V. B. Krishna, G. A. Weaver, and W. H. Sanders, "Pca-based method for detecting integrity attacks on advanced metering infrastructure," in *International Conference on Quantitative Evaluation of Systems*, pp. 70–85, Springer, 2015.

[73] K. G. Lore, D. M. Shila, and L. Ren, "Detecting data integrity attacks on correlated solar farms using multi-layer data driven algorithm," in *2018 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9, IEEE, 2018.

[74] S. Bouktif, A. Fiaz, A. Ouni, and M. Serhani, "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7, p. 1636, 2018.

[75] C. Olah, "Understanding lstm networks," 2015.

[76] M. Sun, C. Feng, E. K. Chartan, B.-M. Hodge, and J. Zhang, "A two-step short-term probabilistic wind forecasting methodology based on predictive distribution optimization," *Applied energy*, vol. 238, pp. 1497–1505, 2019.

[77] J. R. Koza, "Genetic programming," 1997.

[78] M. Sun, C. Feng, and J. Zhang, "Multi-distribution ensemble probabilistic wind power forecasting," *Renewable Energy*, vol. 148, pp. 135–149, 2020.

[79] C. Feng, M. Sun, and J. Zhang, "Reinforced deterministic and probabilistic load forecasting via q-learning dynamic model selection," *IEEE Transactions on Smart Grid*, 2019.

[80] M. Sun, C. Feng, and J. Zhang, "Conditional aggregated probabilistic wind power forecasting based on spatio-temporal correlation," *Applied Energy*, vol. 256, p. 113842, 2019.

[81] H. A. Nielsen, H. Madsen, and T. S. Nielsen, "Using quantile regression to extend an existing wind power forecasting system with probabilistic forecasts," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 9, no. 1-2, pp. 95–108, 2006.

[82] Scikit-learn, "Machine Learning in Python." `https://scikit-learn.org/stable/`.

[83] S. Guha, N. Mishra, G. Roy, and O. Schrijvers, "Robust random cut forest based anomaly detection on streams," in *International conference on machine learning*, pp. 2712–2721, PMLR, 2016.

[84] Z. Zhang, "Introduction to machine learning: k-nearest neighbors," *Annals of translational medicine*, vol. 4, no. 11, 2016.

[85] G. Biau, "Analysis of a random forests model," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 1063–1095, 2012.

[86] Christian Brown, Gregory Von Wald, "Predicting and Explaining Price-Spikes in Real-Time Electricity Markets." `http://cs229.stanford.edu/proj2017/final-reports/5243809.pdf`, 2017.

[87] X. Lu, Z. Y. Dong, and X. Li, "Electricity market price spike forecast with data mining techniques," *Electric power systems research*, vol. 73, no. 1, pp. 19–29, 2005.

[88] N. Amjady and F. Keynia, "Electricity market price spike analysis by a hybrid data model and feature selection technique," *Electric Power Systems Research*, vol. 80, no. 3, pp. 318–327, 2010.

[89] T. D. Mount, Y. Ning, and X. Cai, "Predicting price spikes in electricity markets using a regime-switching model with time-varying parameters," *Energy Economics*, vol. 28, no. 1, pp. 62–80, 2006.

[90] D. Huang, H. Zareipour, W. D. Rosehart, and N. Amjady, "Data mining for electricity price classification and the application to demand-side management," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 808–817, 2012.

[91] T. M. Christensen, A. S. Hurn, and K. A. Lindsay, "Forecasting spikes in electricity prices," *International Journal of Forecasting*, vol. 28, no. 2, pp. 400–411, 2012.

[92] J. H. Zhao, Z. Y. Dong, X. Li, and K. P. Wong, "A framework for electricity price spike analysis with advanced data mining methods," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 376–385, 2007.

[93] R. Weron, "Electricity price forecasting: A review of the state-of-the-art with a look into the future," *International journal of forecasting*, vol. 30, no. 4, pp. 1030–1081, 2014.

[94] UtilityDive, "Pjm, new york electricity markets experience price spikes." `https://www.utilitydive.com/news/washington-dc-power-outage-spikes-pjm-prices/384805/`, 2015.

[95] S. G. Platts. https://www.spglobal.com/platts/en/market-insights/topics/hydrogen, 2018.

[96] S. Stoft, *PJM's capacity market in a price-spike world*. University of California Energy Institute, 2000.

[97] ISO-NE, "Iso new england web services api v1.1." https://webservices.iso-ne.com/docs/v1.1/index.html, 2020.

[98] A. Vidhya, "Complete machine learning guide to parameter tuning in gradient boosting (gbm) in python." https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gradient-boosting-gbm-python/, 2020.

[99] NOAA, "Climate data online." https://www.ncdc.noaa.gov/cdo-web/, 2020.

[100] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "The 2015 ukraine blackout: Implications for false data injection attacks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3317–3318, 2016.

[101] NERC, "Lesson learned: Risks Posted by Firewall Firmware Vulnerability." https://www.nerc.com/pa/rrm/ea/Lessons%20Learned%20Document%20Library/20190901_Risks_Posed_by_Firewall_Firmware_Vulnerabilities.pdf., 2019.

[102] R. Moslemi, A. Mesbahi, and J. M. Velni, "Design of robust profitable false data injection attacks in multi-settlement electricity markets," *IET Generation, Transmission & Distribution*, vol. 12, no. 6, pp. 1263–1270, 2017.

[103] H. Xu, Y. Lin, X. Zhang, and F. Wang, "Power system parameter attack for financial profits in electricity markets," *IEEE Transactions on Smart Grid*, 2020.

[104] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Generalized fdia-based cyber topology attack with application to the australian electricity market trading mechanism," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3820–3829, 2017.

[105] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, "A framework for cyber-topology attacks: Line-switching and new attack scenarios," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1704–1712, 2017.

[106] M. Esmalifalak, H. Nguyen, R. Zheng, L. Xie, L. Song, and Z. Han, "A stealthy attack against electricity market using independent component analysis," *IEEE Systems Journal*, vol. 12, no. 1, pp. 297–307, 2015.

[107] M. R. Mengis and A. Tajer, "Data injection attacks on electricity markets by limited adversaries: Worst-case robustness," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5710–5720, 2017.

[108] A. Tajer, "False data injection attacks in electricity markets by limited adversaries: stochastic robustness," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 128–138, 2017.

[109] H. Ye, Y. Ge, X. Liu, and Z. Li, "Transmission line rating attack in two-settlement electricity markets," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1346–1355, 2015.

[110] D.-H. Choi and L. Xie, "Ramp-induced data attacks on look-ahead dispatch in real-time power markets," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1235–1243, 2013.

[111] C. Liu, M. Zhou, J. Wu, C. Long, and D. Kundur, "Financially motivated fdi on sced in real-time electricity markets: attacks and mitigation," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 1949–1959, 2017.

[112] R. Deng, G. Xiao, and R. Lu, "Defending against false data injection attacks on power system state estimation," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 1, pp. 198–207, 2015.

[113] Y. Huang, J. Tang, Y. Cheng, H. Li, K. A. Campbell, and Z. Han, "Real-time detection of false data injection in smart grid networks: An adaptive cusum method and analysis," *IEEE Systems Journal*, vol. 10, no. 2, pp. 532–543, 2014.

[114] G. Chaojun, P. Jirutitijaroen, and M. Motani, "Detecting false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2476–2483, 2015.

[115] D. Choi and L. Xie, "Sensitivity analysis of real-time locational marginal price to scada sensor data corruption," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1110–1120, 2013.

[116] D.-H. Choi and L. Xie, "Economic impact assessment of topology data attacks with virtual bids," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 512–520, 2016.

[117] N. A. Ruhi, K. Dvijotham, N. Chen, and A. Wierman, "Opportunities for price manipulation by aggregators in electricity markets," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5687–5698, 2017.

[118] Q. Zhang, F. Li, H. Cui, R. Bo, and L. Ren, "Market-level defense against fdia and a new lmp-disguising attack strategy in real-time market operations," *IEEE Transactions on Power Systems*, 2020.

[119] F. Li and R. Bo, "Small test systems for power system economic studies," in *IEEE PES general meeting*, pp. 1–4, IEEE, 2010.

[120] California Independent System Operator, "CAISO Energy Markets Price Performance Report." http://www.caiso.com/Documents/FinalReport-PricePerformanceAnalysis.pdf, 2019.

[121] A. R. Datta and S. Datta, "Electricity market price-spike classification in the smart grid," in *2016 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, 2016.

[122] D. He and W. Chen, "A real-time electricity price forecasting based on the spike clustering analysis," in *2016 IEEE/PES Transmission and Distribution Conference and Exposition (T D)*, pp. 1–5, 2016.

[123] Z. Jiang, J. Wang, T. Zhang, G. Li, and M. Zhou, "Deep learning-based hybrid model for forecasting locational marginal prices," in *2020 IEEE/IAS Industrial and Commercial Power System Asia (I CPS Asia)*, pp. 1733–1738, 2020.

[124] California ISO, "California ISO Oasis Production." http://oasis.caiso.com/, 2020.

[125] T. Kohonen and T. Honkela, "Kohonen network," *Scholarpedia*, vol. 2, no. 1, p. 1568, 2007. revision #127841.

[126] Laboratory of Computer and Information Science, Helsinki University of Technology, "SOM Toolbox." http://www.cis.hut.fi/projects/somtoolbox/, 2020.

[127] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, vol. 37, no. 2, pp. 233–243, 1991.

[128] A. Liaw, M. Wiener, *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[129] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[130] R. Lipovsky and A. Cherepanov, "Blackenergy trojan strikes again: Attacks ukrainian electric power industry." http://www.welivesecurity.com/2016/01/04/blackenergy-trojan-strikes-again-attacks-ukrainian-electric-power-industry/, 2016.

[131] L. Constantin, "New havex malware variants target industrial control systems." http://www.pcworld.com/article/2367240/new-havex-malware-variants-target-industrial-control-system-and-scada-users.html, 2014.

[132] K. Wilhoit and J. Gogolinski, "Sandworm to blacken: The scada connection." http://blog.trendmicro.com/trendlabs-security-intelligence/sandworm-to-blacken-the-scada-connection/, 2014.

[133] ICS-Cert, "Industrial control systems." https://ics-cert.us-cert.gov/, 2014.

[134] Dragos, "North american electric cyber threat perspective." https://www.dragos.com/wp-content/uploads/NA-EL-Threat-Perspective-2019.pdf?hsCtaTracking=761f6e40-d390-4762-9144-cb4ac00ce695%7C5186bf64-df46-47f0-bc02-d5b3424365b8, 2020.

[135] Proofpoint, "Lookback forges ahead: Continued targeting of the united states' utilities sector reveals additional adversary ttps." https://www.proofpoint.com/us/threat-insight/post/lookback-forges-ahead-continued-targeting-united-states-utilities-sector-reveals, 2019.

[136] D. E. Sanger and N. Perlroth, "U.s. escalates online attacks on russia's power grid." https://www.nytimes.com/2019/06/15/us/politics/trump-cyber-russia-grid.html, 2019.

[137] T. Riley, "The cybersecurity 202: Securing the electric grid should be priority for biden's first 100 days, expert says." https://www.washingtonpost.com/politics/2020/12/08/cybersecurity-202-securing-electric-grid-should-be-priority-biden-first-100-days-expert-says/, 2020.

[138] U. S. G. A. Office, "Critical infrastructure protection – actions needed to address significant cybersecurity risks facing the electric grid." https://www.gao.gov/assets/710/701079.pdf, 2019.

[139] U. D. of Energy, "Department of energy's electricity advisory committee establishes the grid resilience for national security subcommittee." https://www.einnews.com/pr_news/531848253/department-of-energy-s-electricity-advisory-committee-establishes-the-grid-resilience-for-national-security-subcommittee, 2020.

[140] I. N. England, "Iso new england proposed 2019 operating and capital budgets." https://www.iso-ne.com/staticassets/documents/2018/09/4_isone_2019_proposed_op_cap_budget_updated_09_25_2018.pdf, 2019.

[141] Dragos, "The dragos platform: Visulize, detect, & respond to ics/ot cybersecurity threats." https://www.dragos.com/platform/, 2021.

[142] Dragos, "Implementing the dragos platform to solve ics cybersecurity challenges in the electric industry." https://www.dragos.com/wp-content/uploads/Dragos-Challenges-In-The-Electric-Industry-Case-Study.pdf?hsCtaTracking=f1686850-b22e-4731-a37a-ccaf83a08b4a%7C239ecc0d-ce3c-4870-a137-010410423555, 2019.

[143] ABB, "Network manager scada/ems and scada/gms." https://new.abb.com/docs/librariesprovider92/enterprise-software/network-manager-scada-ems-gms-overview_abb.pdf, 2021.

[144] Siemens, "Rugged communications equipment for harsh environments." https://new.siemens.com/global/en/products/automation/industrial-communication/rugged-communications.html, 2021.

[145] S. Electric, "Ecostruxure cybersecurity admin expert." https://www.se.com/ww/en/product-range-presentation/63515-ecostruxure%E2%84%A2-cybersecurity-admin-expert/, 2021.

[146] R. Technologies, "Cyber physical systems security." https://www.raytheonintelligenceandspace.com/capabilities/products/cyber-physical, 2021.

[147] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on power systems*, vol. 32, no. 4, pp. 3258–3265, 2016.