Unobservable False Data Injection Attacks on Power Systems

by

Zhigang Chu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved April 2020 by the
Graduate Supervisory Committee:

Oliver Kosut, Chair
Lalitha Sankar
Anna Scaglione
Anamitra Pal

ARIZONA STATE UNIVERSITY

May 2020

ABSTRACT

Reliable operation of modern power systems is ensured by an intelligent cyber layer that monitors and controls the physical system. The data collection and transmission is achieved by the supervisory control and data acquisition (SCADA) system, and data processing is performed by the energy management system (EMS). In the recent decades, the development of phasor measurement units (PMUs) enables wide area real-time monitoring and control. However, both SCADA-based and PMU-based cyber layers are prone to cyber attacks that can impact system operation and lead to severe physical consequences.

This dissertation studies false data injection (FDI) attacks that are unobservable to bad data detectors (BDD). Prior work has shown that an attacker-defender bi-level linear program (ADBLP) can be used to determine the worst-case consequences of FDI attacks aiming to maximize the physical power flow on a target line. However, the results were only demonstrated on small systems assuming that they are operated with DC optimal power flow (OPF). This dissertation is divided into four parts to thoroughly understand the consequences of these attacks as well as develop countermeasures.

The first part focuses on evaluating the vulnerability of large-scale power systems to FDI attacks. The solution technique introduced in prior work to solve the ADBLP is intractable on large-scale systems due to the large number of binary variables. Four new computationally efficient algorithms are presented to solve this problem.

The second part studies vulnerability of $N-1$ reliable power systems operated by state-of-the-art EMSs commonly used in practice, specifically real-time contingency analysis (RTCA), and security-constrained economic dispatch (SCED). An ADBLP is formulated with detailed assumptions on attacker's knowledge and system operations.

The third part considers FDI attacks on PMU measurements that have strong

temporal correlations due to high data rate. It is shown that predictive filters can detect suddenly injected attacks, but not gradually ramping attacks.

The last part proposes a machine learning-based attack detection framework consists of a support vector regression (SVR) load predictor that predicts loads by exploiting both spatial and temporal correlations, and a subsequent support vector machine (SVM) attack detector to determine the existence of attacks.

DEDICATION

This dissertation is dedicated to my wife Ruolei Ji for her endless love and support through all these years.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation and thanks to my advisors, Dr. Oliver Kosut and Dr. Lalitha Sankar, for their guidance, encouragement, and invaluable support throughout my research work and my life as a Ph.D student.

I am also grateful to my committee members, Dr. Anna Scaglione and Dr. Anamitra Pal, for their valuable suggestions and comments.

In addition, I would like to express my gratitude to the National Science Foundation (NSF), Department of Homeland Security (DHS), and Power Systems Engineering Research Center (PSERC) for the financial support provided.

Last but not least, I would like to thank all my friends in the areas of electric power and energy systems, electrical engineering, and industrial engineering, for their compelling intellectual discussion as well as their friendship and support.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

1.1 Overview

With integration of real-time monitoring, sensing, communication and data processing, electric power systems are becoming increasingly efficient and intelligent. This integration is accomplished by the supervisory control and data acquisition (SCADA) systems and energy menagement systems (EMS). SCADA monitors the physical system and collects measurements, which includes voltage and current magnitudes, line power flows, as well as bus power injections, and then send them to control center. However, these measurements are affected by noise, and some of them may be missing when transmitting from sensors to SCADA, or from SCADA to control center. Due to the incompleteness and inaccuracy of the measurements, state estimation (SE) is utilized to estimate the system operating states (voltage magnitudes and angles) from measurements with sufficiently high accuracy. This estimate along with the sub-sequential data processing, optimization and communication, make the real-time control of the power system achievable. Moreover, phasor measurement units (PMUs) have been widely deployed in power systems for monitoring, protection, and control purposes in the past decade. Since PMUs can directly measure the system states with high sampling rate and accuracy, they have the potential to play a significant role in real-time power system SE and dynamic security assessment.

However, similar to all computer network integrated systems, the integration of the cyber layer communication and control also makes power systems more vulnerable to cyber-attacks, which can compromise the measurements, states, topology and

generator dynamics, resulting in serious physical consequences or even system failure. cyber-attacks against the communication and computing infrastructure of the monitoring and control systems of electric power systems have become a growing concern [1, 2]. In recent years, several incidents have shown that cyber-attacks can have severe physical consequences, and power system is vulnerable to these attacks due to lack of detection and protection schemes. Some of these incidents are briefly discussed below, as the motivation of this research.

- In August 2003, a line outage in Ohio was not noticed by the system operator for a sufficient period of time, resulting in a wide area blackout involving parts of the northeastern and midwestern of United States as well as Ontario of Canada, affecting 55 million people. The power was not restored in 4 days until the blackout cost 4 to 10 billion dollars loss [3]. One of the most important reason was identified as "the inadequate situational awareness" [3], which indicates that a similar incident caused by a cyber attack can have severe consequences due to lack of detection and alarm system.

- In 2007, Idaho National Laboratory performed the Aurora test, in which a computer virus intentionally switched a diesel generator's circuit breaker on and off rapidly. As a result, the generator can be out of synchronization and damaged [4]. This tests demonstrated that power system can be overtaken by computer virus that can result in physical consequences.

- In 2012, the Industrial Control Systems Cyber Emergency Response Team revealed that the number of reported cyber attacks was growing and an increasing number of companies who have access to power grid was becoming the cyber attack targets [5]. The U.S. Department of Energy reported that from 2011 to 2014, 362 reports of physical or cyber attacks that interrupted power services

were received from electric utilities [6]. Based on a Department of Homeland Security record, 161 cyber attacks were targeting on the energy sector in 2013, compared to just 31 in 2011 [6], accounting for 60% of all cyber attacks on cyber-physical systems.

- In 2015, a regional power outage took place in Ukraine, which was cause by a third party's illegal entry into the computer and SCADA system of a electricity distribution company, disconnecting 7 substations for 3 hours. Later, three other distribution companies were attacked, resulting in several outages that caused approximately 225 thousand customers to lose power through various area [7]. This attack alarms the power engineers the necessity of defending functionality in power system.

- In 2018, the Department of Homeland Security released a report indicating that Russian operatives have gained access to American electric, nuclear, water, aviation, and critical manufacturing sectors [2]. Though no attacks are actually launched, power engineers should be aware of the potential threats and prepare responses in advance.

Figure 1.1 illustrates the number of DHS recorded cyber incidents on the energy sector in the recent decade [8]. It can be seen that the energy sector is under continuous attack and even though the number reduced after 2013, it could be because the attackers are becoming increasingly intelligent that they are not found by the defense system.

From the list above and figure 1.1, it is obvious that the cyber layer of the power system is vulnerable to cyber attacks, and adversaries are actively attempting to attack the power system. Therefore, it is crucial to evaluate system vulnerability to credible attacks before they happen, and develop techniques to detect potential

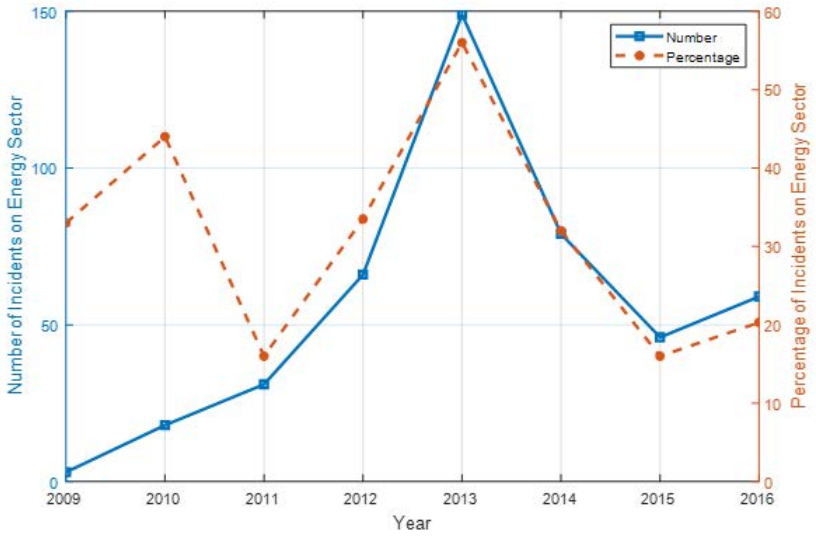

Figure 1.1: DHS Recorded Cyber Incidents on the Energy Sector

attacks and protect the system. Assessing and evaluating consequences of possible attacks is extremely instructive to system operators, and understanding the procedures for potential attacks is important to the secure operation of the power system. Specifically, this dissertation focuses on false data injection (FDI) attacks, which involves a malicious adversary replacing a subset of measurements with counterfeits.

1.2 Literature Review

Cyber security on power systems has gained much interest during the last decade. Since it is impossible to review all of them, this dissertation focuses on unobservable FDI attacks. FDI attacks have been studied for several years to evaluate system vulnerability. It has been shown that FDI attacks can be designed to target system states [9], [10], [11], system topology [12], [13], generator dynamics [14], and energy markets [15]. [9] first introduces a class of FDI attacks on DC SE that is unobservable to the control center. It shows that an attacker with sufficient system knowledge and computational capability can maliciously inject false data without being detected by

existing bad data detection techniques. In [10], the trade-off between attacker's effort to maximize the attack intensity and minimize the detection rate is discussed. [11] studies FDI attacks on AC SE and demonstrates the knowledge needed for such attacks. [15] demonstrates the impacts of FDI attacks on electric power markets, characterized by the change in locational marginal price. [16] introduces unobservable FDI attacks against AC SE that can lead to line overflow. [17] demonstrates that FDI attacks can cooperate with topology attacks to make the on/off status of a line unobservable to control center. Many existing work evaluating the worst-case attack consequences involve solving attacker-defender bi-level linear programs (ADBLPs), wherein the first level models the attacker's objective and limitations (*e.g.*, number of measurements to change), while the second level models the system response under attack via DC optimal power flow (OPF). Examples include attacks that cause line overflows [18], locational marginal price (LMP) changes [19], operating cost increases [20] and sequential outages [21]. The authors of [22] analyzes the physical consequences when the attacker only has limited information, and [23] and [24] focus on cyber-physical coordinated attacks. The authors of [25] propose an ADBLP to find FDI attacks that add or drop contingency pairs with minimum attack effort, and analyze the economic effect of such attacks on LMPs. Rahman *et al.* [26] demonstrate several case studies to showcase the impact of FDI attacks on contingency analysis, but their approach is not optimization-based, which means that it does not consider worst-case scenarios. Both [25] and [26] consider simplified security-constrained economic dispatch (SCED) as system response, but the only addition of their SCED to DCOPF is the contingency case line power flow constraints modeled using DC line outage distribution factors (LODFs), while other SCED constraints such as reserve and ramp rate constraints are not considered.

Techniques to solve ADBLPs with applications to power systems have been stud-

ied in [27, 28], but are limited to scenarios with the same objective for both levels, and hence, their techniques cannot be applied to either the problem in [18] or its generalization for large power systems. An ADBLP can be reformulated as a mathematical program with equilibrium constraints (MPEC) [29] by replacing the second level by its Karush-Kuhn-Tucker (KKT) conditions. However, MPECs are non-convex and hard to solve efficiently in general [30]. Many heuristics have been applied to MPECs involving reformulations and relaxations [31, 32, 33], but they typically require non-linear programs and/or proprietary solvers. Moreover, they do not guarantee optimality, convergence, nor speed. The MPEC from the ADBLP can be further reformulated as a mixed-integer linear program (MILP) by rewriting the complementary slackness constraints as mixed-integer constraints. As the system size increases, this MILP becomes harder to solve due to the increasing number of binary variables. As the system model becomes more complicated, the number of binary variables also increases due to the increasing number of constraints.

As to cyber-attacks on PMUs, the authors of [34, 35] classify the potential cyber-attacks on PMUs as communication link damage attacks, denial of service attacks, data spoofing attacks including GPS spoofing attacks, and FDI attacks. PMU protection and attack detection have gained much interest during the last decade. In [36], Kim and Tong introduce a protection scheme by placing secure PMUs to simultaneously ensure observability and prevent FDI attacks. However, as PMUs are also prone to attacks, their approach cannot thwart FDI attacks when PMU measurements are compromised by attackers. The authors of [37] propose a decentralized FDI attack detection approach based on the Markov graph of bus voltage angles. The drawback of this approach is that it may not perform well when the system experiences a disturbance. An expectation-maximization based detector is introduced by Lee and Kundur in [38] to detect FDI attacks on PMUs, but it only assumes DC power flow model

and requires the bus power injections to be known. Using measurements obtained from deployed PMUs in the grid, [39] and [40] illustrate the low-rank nature of PMU data when it is structured in a matrix. Utilizing this low-rank observation, [41, 42] propose a low-rank decomposition (LRD) based detector to detect FDI attacks on PMU measurements, but [43, 44] propose two different FDI attack schemes that can bypass the LRD detector.

Various attack detection techniques have been presented in the literature. In [45], the authors propose a multivariate Gaussian-based anomaly detector trained using correlation features of micro phasor measurement units (μPMUs), but this detector requires installation of μPMUs in the system. Liu *et al.* [46] detect and identify attacks using reactance perturbation, but this method only works when the attacker has limited resources. The authors of [21] attempt to mitigate FDI attacks using a tri-level optimization approach, and the authors of [47] try to identify FDI attacks by monitoring abnormal load deviations and suspicious branch flow changes. However, they both only focus on attacks that cause line overflows. In [48], a financially motivated FDI attack model is analyzed and a robust incentive-reduction strategy is proposed to deter such attacks by protecting a subset of meters. More generally, machine learning techniques are also deployed in detecting FDI attacks. For example, [49] proposes supervised and semi-supervised machine learning algorithms to detect FDI attacks by exploiting the relationships between statistical and geometric properties of attack vectors employed in the attack scenarios. A deep reinforcement learning-based approach is proposed to detect FDI attacks in [50]. In [51], three machine learning techniques are introduced for attack detection, namely nearest neighbor, semi-supervised one class SVM, and replicator neural network. These three algorithms compare estimated loads with historical loads and use thresholding to determine the existence of FDI attacks.

1.3 Dissertation Objectives

This dissertation focuses on vulnerability analysis of power systems to unobservable FDI attacks and detection techniques. It has been shown in many existing work that these attacks can be designed with an ADBLP and cause physical/economic consequences. They can re-distribute the loads by changing SCADA measurements, to trigger generation re-dispatches that result in physical overflow, cost increase, etc. However, the results are only demonstrated for small systems with tens of buses. In practice, power systems are typically very large. Evaluating the vulnerability of large power systems requires solving large-scale optimization problems within reasonable amount of time, but the optimization problems can be difficult to solve due to the increasing computational burden as the system size scales. Therefore, the first objective of this dissertation is to explore scalable optimization techniques to solve the ADBLP on large-scale power systems, to allow for vulnerability assessment of significantly larger systems (*i.e.* thousands of buses). In particular, we focus on unobservable FDI attacks that aim to maximize the physical power flow on a target line after re-dispatch [18].

Another drawback of the prior work is that they consider only DCOPF as the system response. However, modern power systems typically do not operate with merely DCOPF, but rather operate with more complicated functions including real-time contingency analysis (RTCA) and SCED to ensure $N-1$ reliability. We found in our experiments that the attacks introduced in [18] fail to cause overflows on systems operating with RTCA and SCED. This observation leads to another question: can attacks designed with complete knowledge of operations lead to more consequences? Note that answering this question inherently focuses on very strong attackers, as in general there is no universally adopted formulation of SCED, and we assume the

attacker knows the SCED formulation for the particular system that it is attacking. Our goal of modeling such strong attackers is to understand whether the grid is resilient to such worst-case attacks.

Moreover, as an increasingly important component of this infrastructure, PMUs are also prone to cyber-attacks. Therefore, it is of great importance to evaluate the vulnerability of PMUs against potential cyber-attacks as well as to develop preemptive countermeasures. In the third part of this dissertation, finite impulse response (FIR) predictive filters are proposed to detect FDI attacks on PMU measurements leveraging their high temporal correlations. In addition, gradually ramping FDI attacks that can avoid detection by predictive filters are also proposed.

Finally, the last goal of this dissertation is to develop techniques to effectively detect the attacks as the first step to thwart them. Unobservable FDI attacks alter measurement data, resulting in loads redistributed among the buses, which in turn leads to generation dispatch changes that cause physical/economic consequences. These attacks belong to a broader class of attacks called load redistribution (LR) attacks. Leveraging historical load data that are available to system operators, we develop machine learning-based attack detection framework to determine the existence of LR attacks.

1.4 Outline of Dissertation

The remainder of this dissertation is organized as follows.

Chapter 2 describes the prior work on unobservable line overflow FDI attacks [18], including state estimation, the assumptions on system operation as well as attacker's knowledge and capability, the attack design ADBLP formulation, and the solving technique.

In Chapter 3, four computationally efficient algorithms to solve the attack design

ADBLP are introduced, namely row generation (RG), row and column generation (RCG), cyber-physical difference maximization (DM), and modified Benders' decomposition (MBD). RG reduces the number of binary variables in the MILP converted from the attack design ADBLP by reducing the number of line limit constraints, while RCG does so by fixing the output of non-marginal generators. RG provides the optimal solution, and RCG provides a lower bound on the physical power flow of the target line. DM yields both an upper bound and a lower bound by maximizing the difference between the cyber and physical power flows on the target line. MBD reformulates the ADBLP into a single level optimization problem leveraging duality theory, and then decompose this problem to iteratively solve it. MBD provides a lower bound, but it can be applied to any ADBLP.

In Chapter 4, FDI attack consequences are evaluated on $N-1$ reliable power systems. We showcase that attacks designed without considering EMS operations including RTCA and SCED do not cause the physical consequences intended by the attacker. Given this observation, we propose an ADBLP modeling SCED as the system response, assuming an extremely strong attacker who has perfect knowledge of EMS operations including RTCA and SCED. Simulation results on the synthetic Texas system with 2000 buses show that the resulting attacks can cause post-contingency overflows.

In Chapter 5, we evaluate the vulnerability of PMUs against FDI attacks as well as develop preemptive countermeasures. Two types of predictive filters, namely three-sample quadratic prediction algorithm (TSQPA), and data-driven five-sample predictive (FSP) filter, are tested to detect FDI attacks on PMU measurements utilizing their high temporal correlations. We demonstrate that these predictive filters can be applied to detect FDI attacks if they are suddenly injected into the system. However, attacker can gradually ramp up the magnitude of the attack and avoid detection.

In Chapter 6, we propose an attack detection framework consists of a support vector regression (SVR)-based load predictor and a support vector machine (SVM)-based attack detector. The SVR load predictor is learned from historical data capturing both spatial and temporal correlations between loads in the system. The SVM attack detector leverages loads predicted by the SVR load predictor to determine the existence of attacks. It is trained using randomly generated attacks aiming to maximally explore the attack space, and tested on two intelligently designed attacks, namely line overflow attacks and cost maximization attacks.

Chapter 2

PRIOR WORK: UNOBSERVABLE LINE OVERFLOW FDI ATTACKS

In this chapter we describe the unobservable line overflow FDI attacks proposed in [18]. This involves the simplified power system operation procedure is introduced, as well as the mathematical formulation for SE and the attack model. Throughout, it is assumed that there are n_b buses, n_{br} branches, n_g generators, and n_m measurements in the system.

2.1 Simplified Power System Operation

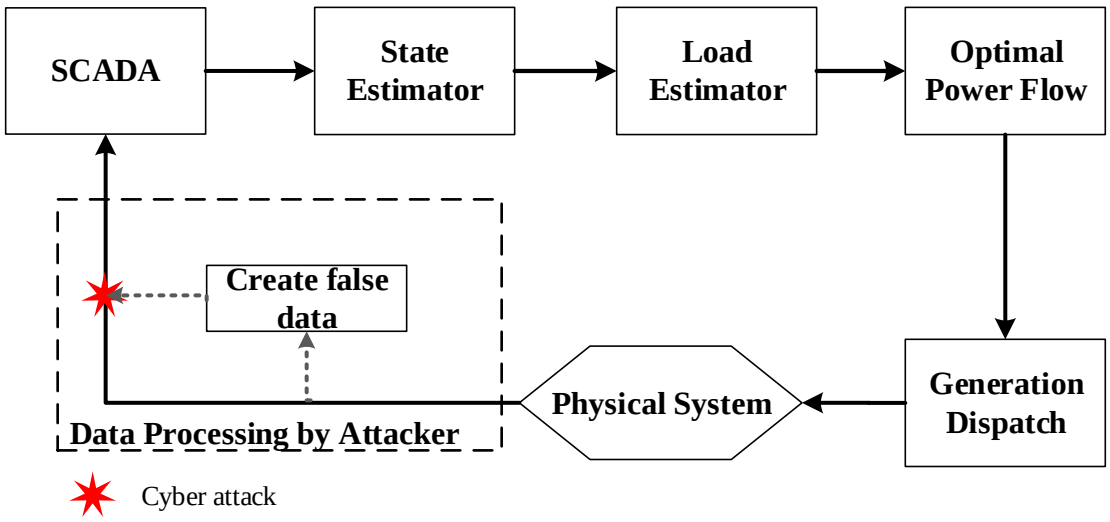

Figure 2.1: Real-time Power System Operation with Attack.

Figure 2.1 demonstrates the simplified power system operation flowchart and how an FDI attack is applied on the system. Without attacks, the power system operates as follows. SCADA collects measurement data from meters and sensors in the physical system, and sends them to state estimator for SE. The state estimator consists of 5

units: topology processor, observability analysis unit, state estimation (SE) unit, bad data detector (BDD), and bad data eliminator. Topology processor determines the current system topology based on breaker status and measurements. Observability analysis is then performed to check whether the system is fully observable, and if it is not, several observable islands will be identified. The system operator then performs SE according to the topology information and measurement data. Bad data detector is utilized to check the existence of bad measurements, and if there exists any bad measurements, the bad data eliminator identifies and filters them. SE is performed again and the solution is achieved until no bad data exists. The SE solution is used to compute the power flow of the system, which subsequently yields the estimated loads of the system. The estimated load information is passed to OPF for optimal generation dispatch. If the state vector is maliciously changed with intelligently designed attack vector, it can result in wrong dispatch that can lead to economic loss, serious physical consequences, or even system failure. For tractability of optimization problems, we focuses on the DC power flow model and DC SE, but the attacks introduced in this research can also be performed against AC SE as in [18].

2.2 State Estimation

The DC measurement model is given by

$$z = H_1 x + e \tag{2.1}$$

Topology information is estimated and observability analysis is performed before the state estimation process to check whether the system is fully observable. The weighted least-square (WLS) method is utilized to solve the SE problem, and the

solution is given by [9]

$$\hat{x} = (H_1^T R^{-1} H_1)^{-1} H_1^T R^{-1} z \tag{2.2}$$

where $\hat{x}$ is the estimated system state vector.

Bad measurements may be introduced to SE due to various reasons such as meter failure, large communication noise, and cyber attacks. Classical BDD detects large errors in measurement data based on measurement residual [52]. The measurement residual is a vector defined as $z - H_1\hat{x}$, which is the difference between observed measurements and estimated measurements. The l_2-norm of the residual is used to detect the existence of any bad data, by comparing with a threshold. This comparison is basiclly a χ^2 test, where the threshold is determined through a hypothesis test with a significance level determined by z. If the l_2-norm of the residual is greater than the threshold, bad data is considered to be present, and measurement with the largest residual will be eliminated. SE is re-run without eliminated measurements to detect and remove any more bad data, until no bad data exists in the measurement vector. Note that traditional bad data detectors cannot necessarily detect FDI attacks. Indeed, unobservable attacks, as defined below, cannot be detected by any bad data detector based on measurement residuals.

2.3 Unobservable Attack Model

To launch unobservable FDI attacks, the attacker is assumed to have the following knowledge and capabilities:

1. The attacker has full system topology information via power transfer distribution factors (PTDF).

2. The attacker has knowledge of load distribution, generation costs, generation limits, and line thermal limits of the system.

3. The attacker has control of the measurements in a subset $\mathcal{S}$ of the network.

As discussed in [18], in the absence of noise, an attack is defined to be *unobservable* when there exists an $n_b \times 1$ attack vector $c \neq 0$ such that for all i, the measurement $\bar{z}_i$ modified by the attacker satisfies $\bar{z}_i = z_i + H_{1i}c$, where H_{1i} denotes the i^{th} row of H_1. Given an attacker with control of the measurements in $\mathcal{S}$, it can execute this attack with attack vector c if $H_{1i}c$ has non-zero entries only in $\mathcal{S}$. Thus, with the limited ability, the attacker can launch an unobservable attack by modifing the measurements using the following rule

$$\bar{z}_i = \begin{cases} z_i\,, & i \notin \mathcal{I}_{\mathcal{S}} \\ z_i + H_{1i}c\,, & i \in \mathcal{I}_{\mathcal{S}} \end{cases} \tag{2.3}$$

where $\mathcal{I}_{\mathcal{S}}$ denotes the set of measurements inside $\mathcal{S}$.

Given an attack vector c, the following procedure produces a subgraph $\mathcal{S}$ that, if controlled by the attacker, can execute an unobservable attack. For an attack vector c, *load buses* (*i.e.*, buses with load) corresponding to non-zero entries of c are denoted as *center buses*. Given an attacker vector c, the subgraph $\mathcal{S}$ controlled by the adversary is constructed using the following algorithm introduced in [11]:

1. Let $\mathcal{S}$ be the set of all center buses.
2. Extend $\mathcal{S}$ by including all branches and buses adjacent to center buses.
3. If any bus on the boundary of $\mathcal{S}$ is a *non-load bus* (*i.e.*, no load is present), extend $\mathcal{S}$ by including all branches and buses adjacent to this bus.
4. Repeat step 3 until all boundary buses are load buses.

Constructing $\mathcal{S}$ with this method ensures that only measurements inside $\mathcal{S}$ can be modified by the attacker. The system operator will see the results of this unobservable

attack as load changes at load buses within $\mathcal{S}$, while the total load of the system remain unchanged.

2.4 DC Optimal Power Flow

DCOPF is a linear approximation of ACOPF around voltage magnitude $V = 1$ and voltage angle $\theta = 0$, neglecting the line resistance and shunt reactance. In contrast to the B-θ method used in [18], DCOPF is formulated using PTDF in this dissertation. This will eliminate the voltage angle variable and have only the generation dispatch P_G as the variable. The DCOPF problem using PTDF is formulated as follows:

$$\underset{P_G}{\text{minimize}} \; C_G\left(P_G\right) \tag{2.4}$$

subject to

$$\textstyle\sum_{g=1}^{n_g} P_{Gg} = \sum_{i=1}^{n_b} P_{Di} \tag{2.5}$$

$$-P^{\max} \leqslant \text{PTDF}(G_B P_G - P_D) \leqslant P^{\max} \tag{2.6}$$

$$P_G^{\min} \leqslant P_G \leqslant P_G^{\max} \tag{2.7}$$

where the variable:

P_G is $n_g \times 1$ vector of generation dispatch

and the parameters:

P_D is the $n_b \times 1$ vector of active load at each bus;

G_B is the $n_b \times n_g$ generator to bus connectivity matrix;

C_G is the cost function of the generation vector;

$P^{\max}$ is the $n_{br} \times 1$ vector of line thermal limits;

$P_G^{\max}, P_G^{\min}$ are $n_g \times 1$ vectors of upper and lower generation limits, respectively;

The objective (2.4) is to minimize the total generation cost. Constraint (2.5) ensures the total system generation equals total load. Constraints (2.6) and (2.7) models the line thermal limits and generation limits, respectively.

2.5 Attack Design ADBLP Formulation

In [18], an FDI attack against state estimation that leads to overflow on a target line is introduced. Subsequent to the attack, the system operator re-dispatches the system generation, leading to an overload on a target line. Modeling such attacks leads to formulation of an ADBLP in which the first level models the attacker's ability and limitations, while the second level models the system response to the attack via DCOPF. The ADBLP is formulated as follows

$$\underset{c}{\text{maximize}} \quad P_l - \sigma \left\|c\right\|_1 \tag{2.8a}$$

subject to

$$P = \text{PTDF}(G_B P_G^* - P_D) \tag{2.8b}$$

$$\left\|c\right\|_1 \leq N_1 \tag{2.8c}$$

$$-L_S P_D \leq Hc \leq L_S P_D \tag{2.8d}$$

$$\{P_G^*\} = \arg\left\{\min_{P_G} C_G\left(P_G\right)\right\} \tag{2.8e}$$

subject to

$$\textstyle\sum_{g=1}^{n_g} P_{Gg} = \sum_{i=1}^{n_b} P_{Di} \qquad (\lambda) \tag{2.8f}$$

$$\begin{aligned} -P^{\max} &\leq \text{PTDF}(G_B P_G - P_D + Hc) \\ &\leq P^{\max} \qquad (F^{\pm}) \end{aligned} \tag{2.8g}$$

$$P_G^{\min} \leq P_G \leq P_G^{\max} \qquad (\alpha^{\pm}) \tag{2.8h}$$

where the variables are:

c attack vector, $n_b \times 1$;

P vector of physical line power flows, $n_{br} \times 1$;

P_l physical power flow of target line l, scalar;

P_G, P_G^* vectors of generation dispatch variables and optimal generation dispatch solved by DCOPF, respectively, both are $n_g \times 1$;

λ dual variable of the load balance constraint;

$F^{\pm}, \alpha^{\pm}$ dual variable vectors of line limits and generation limits, respectively;

and the parameters are:

L_S load shift factor, in percentage;

P_D vector of real loads, $n_b \times 1$;

N_1 l_1-norm limit, scalar;

H dependency matrix between power injection measurements and states, $n_b \times n_b$;

G_B generators to buses connectivity matrix, $n_b \times n_g$;

C_G generation cost vector, $n_g \times 1$;

$P^{\max}$ line limits vector, $n_{br} \times 1$;

$P_G^{\min}, P_G^{\max}$ generation limits vectors, both $n_g \times 1$;

σ penalty of the norm of attack vector c, scalar.

In (2.8a), the penalty factor σ is a small positive number to limit the attack size; constraint (2.8b) is the physical power flow equation; constraint (2.8c) models the attacker's limited resources. Ideally, l_0-norm should be used to precisely capture the sparsity of c, but for tractability reasons we use the l_1-norm as a proxy. Constraint (2.8d) limits the percentage of load changes at each bus to avoid detection. DCOPF (2e)–(2h) models the system response to the attack.

The following modifications can re-formulate the non-linear problem (2.8) into an equivalent MILP as in [18]:

1. Introduce a slack variable s to linearize the l_1-norm constraint in (2.8c) as

$$c \leq s, \quad -c \leq s, \quad \sum_{i \in \mathcal{L}_{\text{load}}} s_i \leq N_1 \tag{2.9}$$

   where $\mathcal{L}_{\text{load}}$ is the set of load buses. This modification simplifies the objective (2.8a) to

$$\underset{c,s}{\text{maximize}} \;\; P_l - \sigma \sum_{i \in \mathcal{L}_{\text{load}}} s_i \tag{2.10}$$

2. Use the KKT optimality conditions (see [53]) to replace the second level DCOPF as constraints (2.8f)–(2.8h), and

$$\begin{aligned} \mathbf{0} = {} & \nabla \left[C_G(P_G) \right] + \nabla \Big(\sum_{g=1}^{n_g} P_{Gg} - \sum_{i=1}^{n_b} P_{Di} \Big) \cdot \lambda \\ & + \nabla \left[\pm \text{PTDF}(G_B P_G - P_D + Hc) - P^{\max} \right] \cdot F^{\pm} \\ & + \nabla \left(P_G - P_G^{\max} \right) \cdot \alpha^{+} + \nabla \left(P_G^{\min} - P_G \right) \cdot \alpha^{-} \end{aligned} \tag{2.11a}$$

$$\mathbf{0} \leq F^{\pm}, \alpha^{\pm} \tag{2.11b}$$

$$\mathbf{0} = \text{diag}(F^{\pm}) \left[\text{PTDF}(G_B P_G - P_D + Hc) \mp P^{\max} \right] \tag{2.11c}$$

$$\mathbf{0} = \text{diag}(\alpha^+)\,(P_G - P_G^{\max}) \tag{2.11d}$$

$$\mathbf{0} = \text{diag}(\alpha^-)\,\left(P_G^{\min} - P_G\right) \tag{2.11e}$$

where constraint (2.11a) is the partial gradient optimal condition, (2.11b) is the dual feasibility constraint, and (2.11c)–(2.11e) are the complementary slackness conditions.

3. Introduce binary variables $\delta_F^{\pm}, \delta_\alpha^{\pm}$ and a large constant M to linearize the complementary slackness conditions as

$$\delta_F^{\pm} \in \{0,1\}^{n_{br}}, \delta_\alpha^{\pm} \in \{0,1\}^{n_g} \tag{2.12a}$$

$$\begin{cases} F^{\pm} \leq M\delta_F^{\pm} \\ P^{\max} \mp \text{PTDF}(G_B P_G - P_D + Hc) \leq M(\mathbf{1} - \delta_F^{\pm}) \end{cases} \tag{2.12b}$$

$$\begin{cases} \alpha^{\pm} \leq M\delta_\alpha^{\pm} \\ P_G^{\max} - P_G \leq M(\mathbf{1} - \delta_\alpha^{+}) \\ P_G - P_G^{\min} \leq M(\mathbf{1} - \delta_\alpha^{-}). \end{cases} \tag{2.12c}$$

The full problem is then converted to a single level MILP with objective (2.10), and constraints (2.8b), (2.8d), (2.8f)–(2.9), (2.11a)–(2.11b), and (2.12). Henceforth, we refer to this optimization problem as *the original MILP* with P_l^* as its optimal objective value. As MILPs are in general NP-hard, guarantees on polynomial time solutions cannot be provided. Note that in the last modification, $2(n_{br} + n_g)$ binary variables are introduced to the original MILP. As the system network size scales, the number of lines and generators in the system increases rapidly due to the high interconnection level of the system. Hence, the number of binary variables also increases rapidly, resulting in an increased computational burden. It has been found experimentally that on the IEEE 118-bus system, the original MILP fails to converge in a reasonable length of time using solver GUROBI, as demonstrated in Table 2.1.

Table 2.1: Computational Efficiency Comparison

Test System	**24-bus**	**118-bus**	**Polish (2383-bus)**
# of binary variables	142	480	6446
Average solving time	5s	>24h	>24h

We have found experimentally that it fails to find a solution in a reasonable length of time on the IEEE 118-bus system using solver GUROBI, as demonstrated in Table 2.1. On the Polish system with even more binary variables, it is expected to take even longer to solve. Applying this optimization problem directly to evaluate the vulnerability of actual systems with thousands of buses to worst-case FDI attacks is intractable.

Chapter 3

COMPUTATIONALLY EFFICIENT ALGORITHMS TO SOLVE ATTACK OPTIMIZATION PROBLEMS

In this section, we introduce four computationally efficient algorithms to solve the attack optimization problem on large-scale power systems. The first algorithm is denoted *row generation for line limit constraints* (RG). This algorithm identifies and eliminates extraneous line limit constraints, reducing the number of binary variables in the MILP. If it converges, the optimal objective $P_l^{*(\mathrm{RG})}$ solved with RG is guaranteed to be equal to the optimal objective of the original MILP. However, as the system size scales, even though the number of binary variables associated with line limit constraints is significantly reduced, the number of binary variables associated with generation limit constraints is still large enough to make the problem hard to solve. We have found experimentally that RG works efficiently for the IEEE 118-bus system, but it fails to converge in a reasonable length of time for the Polish system with 2383 buses.

Thus, similar to RG, we introduce our second algorithm, denoted *row and column generation for line and generator limit constraints* (RCG), in which we further reduce the number of binary variables by eliminating generation limit constraints in addition to line limit constraints. RCG gives a feasible solution, but the solution can be sub-optimal. Thus, the resulting objective value $P_l^{*(\mathrm{RCG})}$ is a lower bound on P_l^*.

Our third algorithm, denoted *cyber-physical-difference maximization* (DM), evaluates system vulnerability without directly modeling the system response. This reduces to a linear program, whose optimal solution can be used to derive both a lower bound $P_l^{*(\mathrm{DM,lb})}$ and an upper bound $P_l^{*(\mathrm{DM,ub})}$ on P_l^*.

Finally, our fourth algorithm is denoted *modified Benders' decomposition for bi-level linear programs* (MBD), which iteratively solves the original bi-level problem without converting it to an MILP. Due to the non-convexity of the bi-level optimization problem, MBD gives a feasible solution, whose corresponding objective value $P_l^{*(\mathrm{MBD})}$ is a lower bound on P_l^*. We proceed to introduce each algorithm in detail.

3.1 Row Generation for Line Limit Constraints

Row and column generation techniques are useful in solving large-scale linear programs. For constraints of the form $Ax \leq b$, row generation retains only a subset of constraints (rows of A), and column generation retains only a subset of variables (columns of A). We iteratively add only those constraints and variables that are needed [54] [55]. These techniques help reduce the size of matrix A, and hence accelerate the solving process. Similar techniques have been used by power system operators for large-scale optimization problems, including unit commitment and security constrained economic dispatch (SCED) [56]. In our problem, these techniques allow us to reduce the number of binary variables.

The original MILP can be solved with less number of $\delta_F^{\pm}$ by modeling only *critical lines* (*i.e.*, lines operating at over 90% of their ratings). If the cyber power flow of a line is beyond its limit, we say this line has *cyber overflow*. If there are any post-attack cyber overflows, the line limit constraints for those lines are added back to the attack optimization problem (new rows generated). If this algorithm terminates, the solution is guaranteed to be optimal (*i.e.* $P_l^{*(\mathrm{RG})} = P_l^*$) because no constraints are violated.

Algorithm 1 Row generation for line limit constraints (RG)

1. Perform DCOPF for the whole system with no attack.
2. Let $\mathcal{Q}$ be the set of critical lines.
3. Remove the constraints corresponding to line k from (2.8g), (2.11a)–(2.11b), (2.12a), and (2.12b) for all $k \notin \mathcal{Q}$.
4. Solve the reduced problem, and compute the cyber power flow of the system using the optimal dispatch $P_G^{*(\mathrm{RG})}$.
5. If cyber overflows exist, add all lines with cyber overflow to $\mathcal{Q}$, and go back to 3).
6. Let $P_l^{*(\mathrm{RG})}$ be the optimal objective value of RG.

3.2 Row and Column Generation for Line and Generator Limit Constraints

RCG further reduces the number of binary variables by reducing generator limits. Since load changes are limited by constraint (2.8d), it is likely that in response to these load changes, only a subset of all generators (denote as *marginal generators*) will re-dispatch. RCG removes the generator limits of non-marginal generators and treats their outputs as constants. In addition to the re-included line limits in RG, non-marginal generators with changed dispatch after the attack are added to the set of marginal generators (new columns generated). This ensures the system response to the attack predicted by the attacker is correct.

Since it does not search the full feasible space by holding some entries of P_G as constants, RCG is not guaranteed to yield the optimal solution for the original MILP. However, it always provides a feasible solution, and hence, $P_l^{*(\mathrm{RCG})}$ is a lower bound

Algorithm 2 Row and column generation for line and generator limit constraints (RCG)

1. Perform DCOPF for the whole system with no attack.
2. Let $\mathcal{Q}$ be the set of critical lines as defined in RG.
3. Let $\mathcal{R}$ be the set of generators g where $P_{Gg}^{\min} < P_{Gg} < P_{Gg}^{\max}$.
4. Remove the constraints corresponding to line k and generator g from (2.8g)–(2.8h), (2.11a) and (2.12a)–(2.12c) for all $k \notin \mathcal{Q}$ and all $g \notin \mathcal{R}$.
5. Solve the reduced problem to find the resulting generation dispatch $P_G^{*(\mathrm{RCG})}$ and optimal attack vector c_{RCG}^*.
6. Use c_{RCG}^* to run post-attack DCOPF (2.8e)–(2.8h) to find system operator's corresponding dispatch P_G^{post}.
7. For all g that $P_{Gg}^{*(\mathrm{RCG})} \neq P_{Gg}^{\mathrm{post}}$, if they belong to $\mathcal{R}$, go to 8); Otherwise add them to $\mathcal{R}$ and go back to 4).
8. Use $P_G^{*(\mathrm{RCG})}$ to calculate the cyber power flows.
9. If cyber overflows exist, add all lines with cyber overflow to $\mathcal{Q}$, and go back to 4).
10. Let $P_l^{*(\mathrm{RCG})}$ be the optimal objective value of RCG.

on P_l^*.

3.3 Cyber-Physical-Difference Maximization

The DM algorithm maximizes the post-attack power flow difference between physical and cyber power flows. Both lower and upper bounds on P_l^* can be derived using DM. Moreover, this algorithm can be applied efficiently on large systems as it only involves solving an LP.

Algorithm 3 Cyber-physical-difference maximization (DM)

1. Solve the following optimization problem and let c^*_{DM} be the optimal solution.

$$\begin{aligned} \underset{c,s}{\text{maximize}} \quad & -\text{PTDF}_l(Hc) \\ \text{subject to} \quad & (2.8\text{b}), (2.8\text{d}), (2.9). \end{aligned} \tag{3.1}$$

2. Obtain the upper bound:

$$P_l^{*(\text{DM,ub})} := P_l^{\max} - \text{PTDF}_l(Hc^*_{\text{DM}}) \tag{3.2}$$

   where PTDF_l is the l^{th} row of the PTDF matrix.

3. Perform post-attack DCOPF (2.8e)–(2.8h) with $c = c^*_{\text{DM}}$ to find post-attack dispatch P_G^{post}.

4. Obtain the lower bound:

$$P_l^{*(\text{DM,lb})} := \text{PTDF}_l(G_B P_G^{\text{post}} - P_D). \tag{3.3}$$

The following theorem proves the DM algorithm.

Theorem 1. *The upper and lower bounds from the DM algorithm satisfy*

$$P_l^{*(DM,lb)} \leq P_l^* \leq P_l^{*(DM,ub)}. \tag{3.4}$$

Moreover, if the post-attack cyber power flow of the target line is at its limit, then $P_l^{*(DM,ub)} = P_l^{*(DM,lb)} = P_l^*$.

Proof. The post-attack physical power flow on target line l is given by

$$P_l^{\text{physical}} = \text{PTDF}_l(G_B P_G^{\text{post}} - P_D). \tag{3.5}$$

This is a feasible solution, and hence, is a lower bound on P_l^*. Given an attack vector c, the post-attack cyber power flow on l is given by

$$\begin{aligned} P_l^{\text{cyber}} &= \text{PTDF}_l(G_B P_G^{\text{post}} - P_D + Hc) \\ &= P_l^{\text{physical}} + \text{PTDF}_l(Hc). \end{aligned} \tag{3.6}$$

Since the cyber power flow cannot exceed its limit,

$$P_l^{\text{cyber}} = P_l^{\text{physical}} + \text{PTDF}_l(Hc) \leq P_l^{\text{max}}. \tag{3.7}$$

Thus, we have

$$P_l^{\text{physical}} \leq P_l^{\text{max}} - \text{PTDF}_l(Hc). \tag{3.8}$$

Therefore, since the optimization problem in (3.1) maximizes the second term in (3.8), $P_l^{*(\text{DM,ub})}$ as computed in (3.2) is an upper bound on P_l^*. If $P_l^{\text{cyber}} = P_l^{\text{max}}$, substituting this relationship into (3.5) and (3.6) proves $P_l^{*(\text{DM,ub})} = P_l^{*(\text{DM,lb})} = P_l^*$. □

3.4 Modified Benders' Decomposition Algorithm to Solve ADBLPs

ADBLPs with different objectives in the two levels are in general non-convex. The authors of [18] solve their ADBLP by replacing the second level defender's problem by its KKT conditions and then convert the problem into an MILP, but this approach does not apply to large-scale systems due to the numerical difficulty brought on by large number of binary variables. To the best of our knowledge, there are no existing techniques to solve large-scale ADBLPs efficiently. In this section, we introduce

a modified Benders' decomposition (MBD) algorithm to solve ADBLPs. Benders' decomposition [57] is an iterative approach to solve linear programs in a distributed manner [58]. It is a popular technique to solve optimization problems of large size or with complicating variables. It is also effective in solving complex optimization problems such as stochastic programs and mixed-integer linear programs. In Benders' decomposition, an optimization problem is decomposed into two sub-problems, wherein variables of each sub-problem are treated as constant in the other. The two sub-problems are solved iteratively until the solution converges. Our MBD algorithm modifies the classic Benders' decomposition algorithm to apply it on any ADBLP.

An ADBLP takes the following form (dual variable of the defender's problem is in parentheses):

$$\underset{u}{\text{minimize}}\ c_1^T u + d_1^T v^* \tag{3.9a}$$

subject to

$$A_1 u \geq b_1 \tag{3.9b}$$

$$v^* = \arg\{\min_{v}\ d_2^T v\} \tag{3.9c}$$

subject to

$$A_2 u + A_3 v \geq b_2 \qquad (\beta) \tag{3.9d}$$

where u and v are the attacker's and defender's decision variables, respectively. The defender has no control on u, and hence, u in (3.9d) is treated as a constant in the defender's problem. The attacker does not directly control v, but it controls v^* by changing u, assuming it has knowledge of the defender's objective and constraints.

The attack optimization ADBLP (4.1) fits in the form of (3.9) where the attack vector c is represented by u and SCED variables P_G, R_G, P, and P_k are represented by v. In the attacker's objective function, $c_1^T u$ represents the term $-\sigma \left\|c\right\|_1$, and $d_1^T v^*$ represents the term P_{l,k_t} in (2.8a). Equality constraints can be equivalently written

as two inequality constraints. For example, (2.8f) can be written as

$$\mathbf{1}^T P_G \geq \mathbf{1}^T P_D \tag{3.10a}$$

$$-\mathbf{1}^T P_G \geq -\mathbf{1}^T P_D \tag{3.10b}$$

which fits the form of (3.9d). One can similarly map all the constraints in (4.1) to those in (3.9).

The defender's problem (3.9c)–(3.9d), which represents the system response (SCED) to a fixed attack vector, has the following dual problem (note that u is treated as constant here since it is the fixed attack vector from the attacker's problem):

$$\underset{\beta}{\text{maximize}}\ \beta^T(b_2 - A_2 u) \tag{3.11a}$$

$$\text{subject to } A_3^T \beta = d_2 \tag{3.11b}$$

$$\beta \geq 0. \tag{3.11c}$$

By weak duality [53], for any feasible primal/dual pair, the dual objective value is always less than the primal one:

$$\beta^T(b_2 - A_2 u) \leq d_2^T v. \tag{3.12}$$

Since the defender's problem is a linear program, it satisfies strong duality. That is, any feasible point (v, β) that satisfies

$$\beta^T(b_2 - A_2 u) \geq d_2^T v \tag{3.13}$$

is an optimal solution to it. Therefore, constraints (3.9d), (3.11b), (3.11c), and (3.13) guarantee the optimality of the defender's problem, and hence, can be used to convert the ADBLP to a single level problem as:

$$\underset{u,v,\beta}{\text{minimize}}\ c_1^T u + d_1^T v \tag{3.14a}$$

$$\text{subject to } A_1 u \geq b_1 \tag{3.14b}$$

$$A_2u + A_3v \geq b_2 \tag{3.14c}$$

$$A_3^T\beta = d_2 \tag{3.14d}$$

$$\beta^T b_2 - \beta^T A_2 u - d_2^T v \geq 0 \tag{3.14e}$$

$$\beta \geq 0. \tag{3.14f}$$

The bilinear term $\beta^T A_2 u$ in (3.14e) is non-convex and hard to solve. To overcome this difficulty, Benders' decomposition is utilized to decompose this optimization problem into two problems, with u as the variable for the master problem (MP) and v, β as the variables for the slave problem (SP). The MP takes the following form:

$$\underset{u,\alpha}{\text{minimize}}\ c_1^T u + \alpha \tag{3.15a}$$

$$\text{subject to } A_1 u \geq b_1 \tag{3.15b}$$

where α is a variable introduced to represent $d_1^T v$, which will then be updated by adding cuts. The SP is given by:

$$\underset{v,\beta}{\text{minimize}}\ d_1^T v \tag{3.16a}$$

$$\text{subject to } \beta^T b_2 - d_2^T v - \beta^T A_2 u \geq 0 \quad (\delta) \tag{3.16b}$$

$$A_3 v \geq b_2 - A_2 u \qquad (\gamma) \tag{3.16c}$$

$$A_3^T \beta = d_2 \qquad (\lambda) \tag{3.16d}$$

$$\beta \geq 0. \tag{3.16e}$$

At the optimal solution of the SP given by (3.16), we have

$$d_1^T v^* = \gamma^T b_2 + \lambda^T d_2 - \gamma^T A_2 u. \tag{3.17}$$

An optimality cut can be added to the MP by taking the right hand side of (3.17):

$$\alpha \geq \gamma^T b_2 + \lambda^T d_2 - \gamma^T A_2 u. \tag{3.18}$$

Note that (3.18) is in the MP, and therefore, u is again a variable. If the SP is infeasible with a given u, slack variables s_i, $i = 1, 2, 3$, can be introduced to all of the SP constraints to solve the relaxed SP:

$$\underset{v,\beta,s_i}{\text{minimize}} \; d_1^T v \tag{3.19a}$$

$$\text{subject to } \beta^T b_2 - d_2^T v - \beta^T A_2 u + s_1 \geq 0 \quad (\hat{\delta}) \tag{3.19b}$$

$$A_3 v + s_2 \geq b_2 - A_2 u \qquad (\hat{\gamma}) \tag{3.19c}$$

$$A_3^T \beta + s_3 = d_2 \qquad (\hat{\lambda}) \tag{3.19d}$$

$$\beta \geq 0. \tag{3.19e}$$

where s_i, $i = 1, 2, 3$ are the slack variables introduced to ensure feasibility of the relaxed SP. Then, instead of an optimality cut (3.18), a feasibility cut is added to the MP:

$$0 \geq \hat{\gamma}^T b_2 + \hat{\lambda}^T d_2 - \hat{\gamma}^T A_2 u. \tag{3.20}$$

The MP and SP can then be solved iteratively, with the MP updating u and the SP updating cuts in each iteration.

Solving the SP is equivalent to solving the second level SCED under attack (2.8e)−(4.1o), while the dual variables of the SP provide information on the objective function (2.8a). Since each cut is formulated linearly on the u domain, adding cuts to the MP does not affect its convexity. Thus, MBD is guaranteed to converge in a finite number of iterations [59]. However, due to the non-convexity of the original bi-level optimization problem, global optimal solution cannot be guaranteed [60]. Therefore, the optimal objective value obtained by MBD, $\hat{P}^*_{l,k_t}$, is a lower bound on P^*_{l,k_t}, the global optimal objective.

Algorithm 4 Modified Benders' Decomposition for Bi-level Linear Programs (MBD)

1. Set the iteration number $j = 1$ and let $u^{(0)} = 0$.
2. Solve the SP (3.16) with $u = u^{(j-1)}$.
3. If the SP is infeasible, solve the relaxed SP (3.19) and obtain $(\hat{\gamma}^{(j)}, \hat{\lambda}^{(j)})$, then add a feasibility cut of form (3.20) to the MP. Otherwise, solve SP (3.16) to get $(v^{(j)}, \beta^{(j)}, \gamma^{(j)}, \lambda^{(j)})$, and add an optimality cut of form (3.18) to the MP.
4. Solve the MP with added cuts and obtain the solution $(u^{(j)}, \alpha^{(j)})$.
5. If $|\frac{d_1^T v^{(j)} - \alpha^{(j)}}{\alpha^{(j)}}| < \epsilon$, stop. The optimal objective value is obtained as $c_1^T u^{(j)} + d_1^T v^{(j)}$. Otherwise, let $j = j + 1$ and go to step 2).

3.5 Simulation Results

In this section, we present numerical results using the algorithms described in Sec. 3.1 - **??**. Two test systems are used, namely the IEEE 118-bus system and the Polish system with 2383 buses. The topologies of the IEEE 118-bus system and the Polish system are shown in Figure 3.1 and 3.2, respectively. Before attack, the IEEE 118-bus system and the Polish system have 7 and 17 critical lines, and 15 and 6 marginal generators, respectively. We exhaustively target all critical lines to assess the vulnerability of these two systems. The l_1-norm limit N_1 is chosen with increment 0.1 in the range $[0.1, 1]$ for the 118-bus system, and $[0.1, 2]$ for the Polish system. Throughout, Matlab, Matpower, and the Gurobi solver are used to perform the simulations. All tests are conducted using a 3.40 GHz PC with 32 GB RAM.

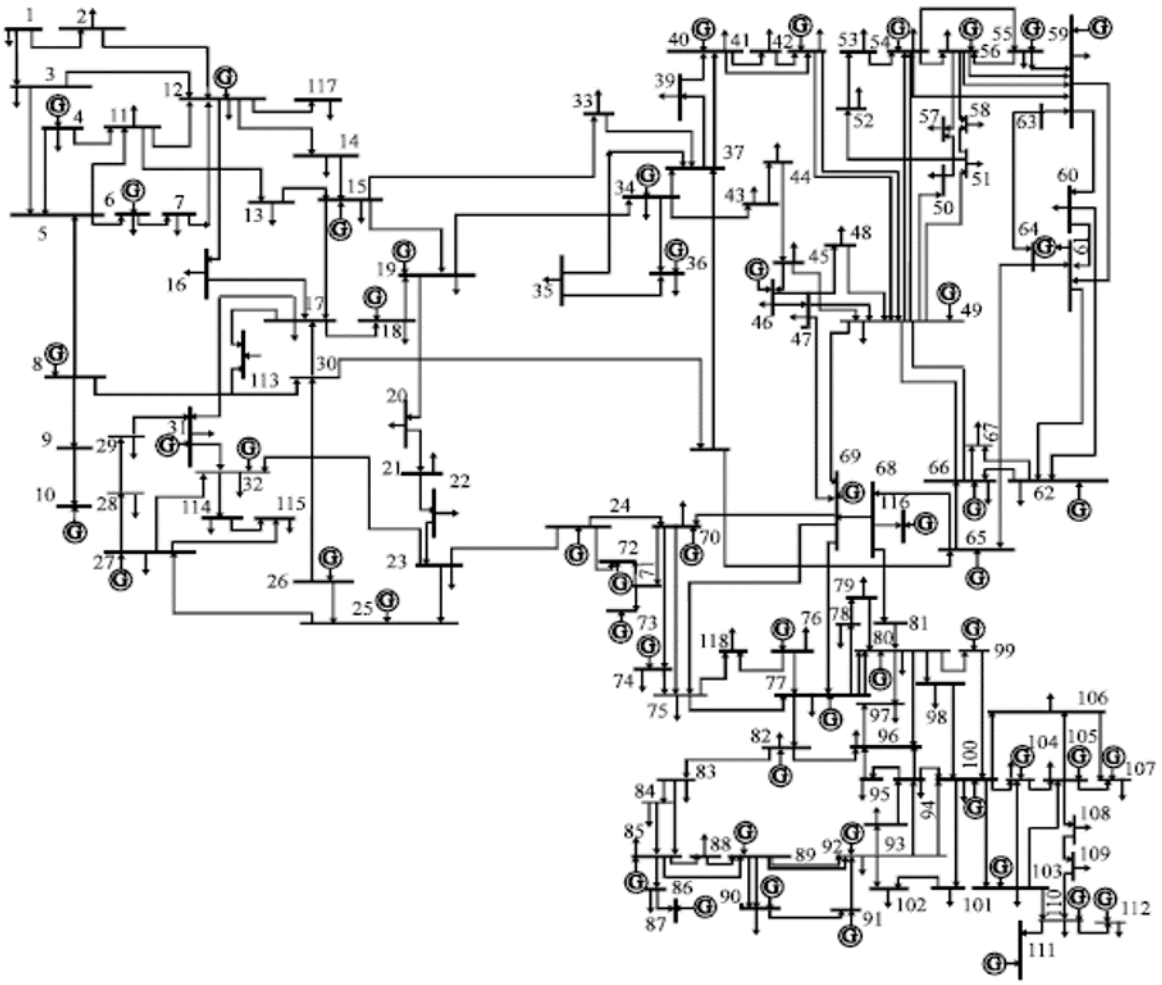

Figure 3.1: IEEE 118-bus System Topology

3.5.1 Computational Efficiency

The decrease in the number of binary variables characterizes the computational efficiency improved by RG and RCG. Table 3.1 illustrates a comparison of the average number of binary variables when applying the original MILP, RG, and RCG. Since we are unable to verify the convergence of RG for the Polish system, the number of binary variables on RG for this system is an estimate. This table demonstrates that both RG and RCG can greatly reduce the number of binary variables compared to the original MILP, and therefore significantly improve the computational efficiency.

Figure 3.2: Polish System Topology

Table 3.1: Comparison of the Average Number of Binary Variables

Test System	Original MILP	RG	RCG
118-Bus	480	122	45
Polish	6446	688	87

Table 3.2 illustrates the statistics of the computation time for several target lines using the proposed algorithms with 10% load shift. For each target line, each algorithm is tested for the full range of N_1 values stated above. We note that RCG is more efficient than RG since it requires fewer binary variables. DM is the most efficient algorithm as it only involves solving an LP. Note that the number of iterations for MBD varies for different parameter choices (target line, N_1, and L_S), resulting in a large variation in computation time.

3.5.2 Results on Maximal Physical Power Flows

Fig. 3.3 illustrates the maximal physical power flows with $L_S = 10\%$ on target lines 104 and 141 of the IEEE 118-bus system. It demonstrates a comparison of the bounds found by RCG, DM and MBD to the optimal solution provided by RG.

Note that for target line 104 with any N_1, all four algorithms yield the optimal solution. For target line 141, we see that $P_l^{*(\mathrm{MBD})} < P_l^{*(\mathrm{DM,lb})} < P_l^{*(\mathrm{RG})} = P_l^{*(\mathrm{RCG})} < P_l^{*(\mathrm{DM,ub})}$, illustrating that $P_l^{*(\mathrm{DM,lb})}$ and $P_l^{*(\mathrm{DM,ub})}$ are not always tight bounds on P_l^*. RCG provides the optimal solution for all target lines we have considered in the 118-bus system.

The maximal power flows with 10% load shift for target lines 292, 24, and 1816 of the Polish system are illustrated in Fig. 3.4. Note that RG is intractable on the Polish system. For target line 292, all three algorithms yield the optimal solution in

Table 3.2: Statistics of Computation Time with 10% Load Shift

Target line	Algorithm	Max (s)	Min (s)	Avg (s)	Med (s)
37 of 118-bus	RG	7.53	0.95	3.33	1.9
	RCG	1.25	0.34	0.76	0.69
	DM	0.5	0.43	0.47	0.45
	MBD	1.88	1.57	1.63	1.59
24 of Polish	RCG	46.36	3.40	20.39	13.67
	DM	15.75	1.91	8.09	8.58
	MBD	12.26	10.46	11.40	11.58
292 of Polish	RCG	76.34	27.47	39.29	33.69
	DM	16.77	1.91	7.02	6.10
	MBD	1846.2	9.86	358.73	10.31

the range $N_1 \in [0.1, 1.6]$, *i.e.,* $P_l^{*(\mathrm{DM,ub})} = P_l^{*(\mathrm{DM,lb})} = P_l^{*(\mathrm{RCG})} = P_l^{*(\mathrm{MBD})}$, but not for the remaining N_1. For target line 24, MBD yields the tightest lower bound; while for target line 1816, DM provides the tightest lower bound.

3.5.3 Results on Attack Resources

Fig. 3.5 illustrates the relationship between maximal power flow and l_0-norm of the attack vector (*i.e.* the number of center buses in the attack) versus the l_1-norm constraint N_1 for target line 292 of the Polish system, with different load shift constraints. As N_1 increases, so does the l_0-norm of the attack, indicating that l_1-norm is a valid proxy for l_0-norm for our problem. If a larger load shift is allowed,

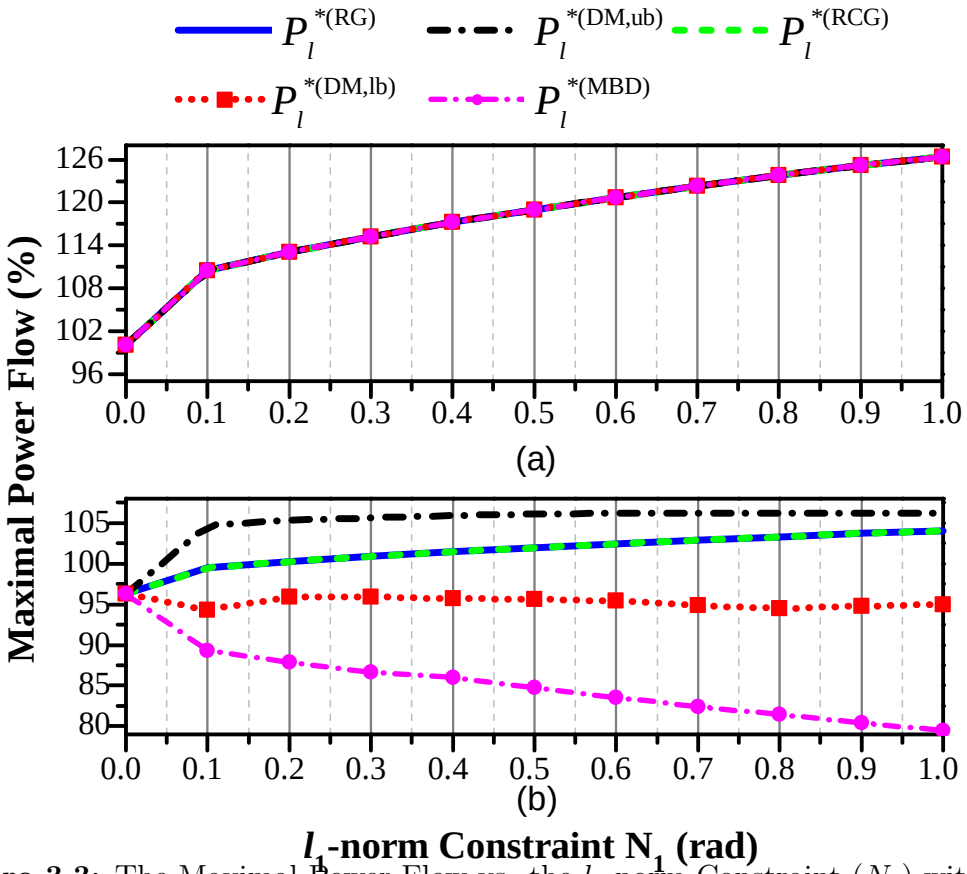

Figure 3.3: The Maximal Power Flow vs. the l_1-norm Constraint (N_1) with Target Line (a) 104, and (b) 141 of IEEE 118-bus System. L_S=10%.

the maximal power flow on target line increases, but the resulting l_0-norm decreases. This indicates a trade-off between load shift and attacker's resources: as the attacker attempts to avoid detection by minimizing load changes, it will require control over a larger portion of the system to launch a comparable attack. Similar results are also obtained on the IEEE 118-bus system.

3.5.4 Line Vulnerability

Since the objective of the attack is to maximize the physical power flow on a target line, it is intuitive that congested lines are more vulnerable to this attack. We have found experimentally that almost every congested line can be overloaded. One

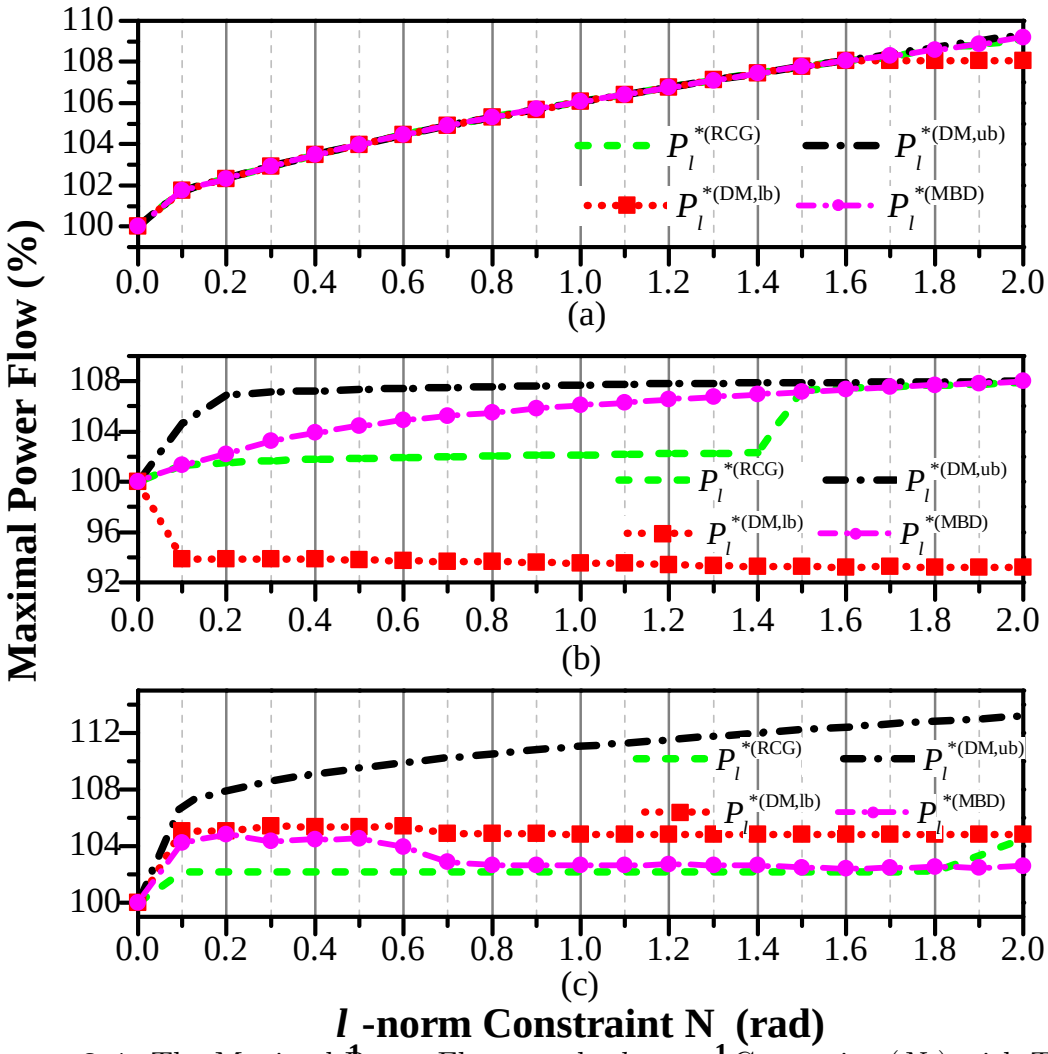

Figure 3.4: The Maximal Power Flow vs. the l_1-norm Constraint (N_1) with Target Line (a) 292, (b) 24, and (c) 1816 of the Polish System. L_S=10%.

exception is line 176 in the IEEE 118-bus system. This is because line 176 is a radial line: it is the only line connected to a bus with a generator and no load. The line limit constraint in the OPF (2.8g) ensures that no possible dispatch could cause the line power flow to exceed the limit, even if based on counterfeit loads. In fact, any line with this radial configuration is immune to the proposed attack; moreover, these radial lines represent the only exceptions to our finding that congested lines can be overloaded. We have also found that lines that are not congested pre-attack may still be vulnerable to this attack, such as line 141 in the IEEE 118-bus system (Fig.3.3(b))

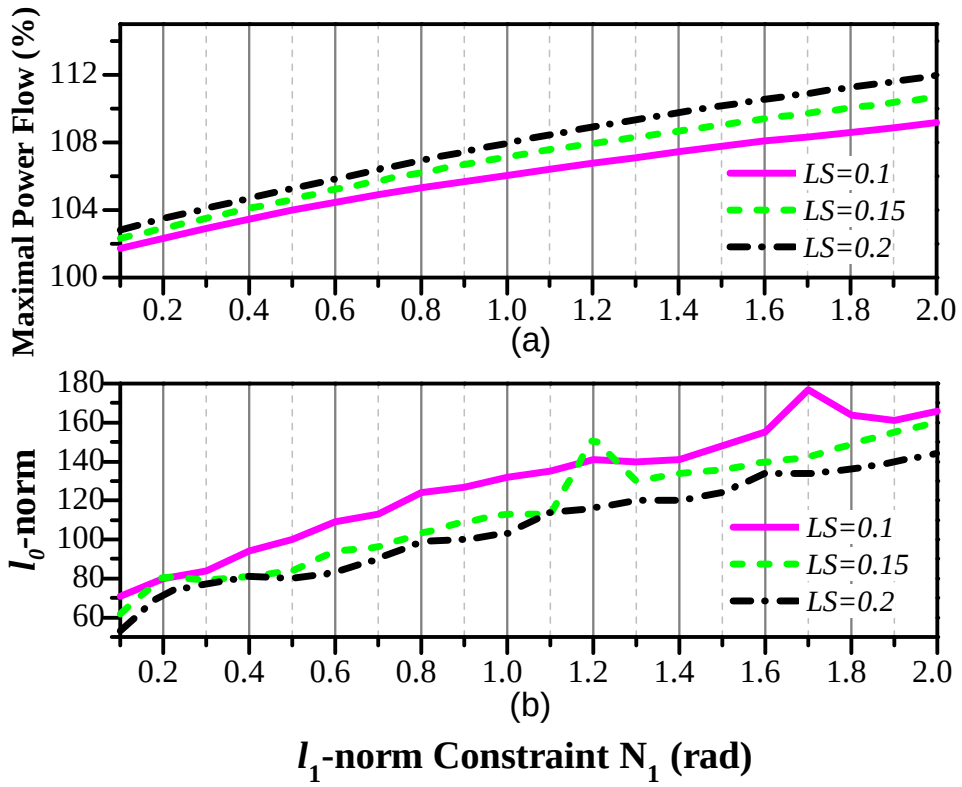

Figure 3.5: (a) The Maximal Power Flow and (b) l_0-norm of the Attack Vector vs. the l_1-norm Constraint (N_1) for Target Line 292 of the Polish System with Different Load Shift.

and line 2110 in the Polish system (Fig. 3.6).

3.5.5 Impact of Overall Congestion on Vulnerability

In the above, we have shown that virtually all critical or congested lines are vulnerable to overload. However, the extent of the vulnerability depends on several factors, such as the overall congestion of the system. This phenomenon is illustrated in Fig. 5, which shows the worst-case attack for line 292 of the Polish system under different overall congestion levels. This overall congestion is adjusted by uniformly changing the line ratings for all lines. Note that higher line ratings mean a less congested system. As shown in Fig. 5, as the overall congestion level increases, the maximal power flow on the target line also increases, even though the line is equally

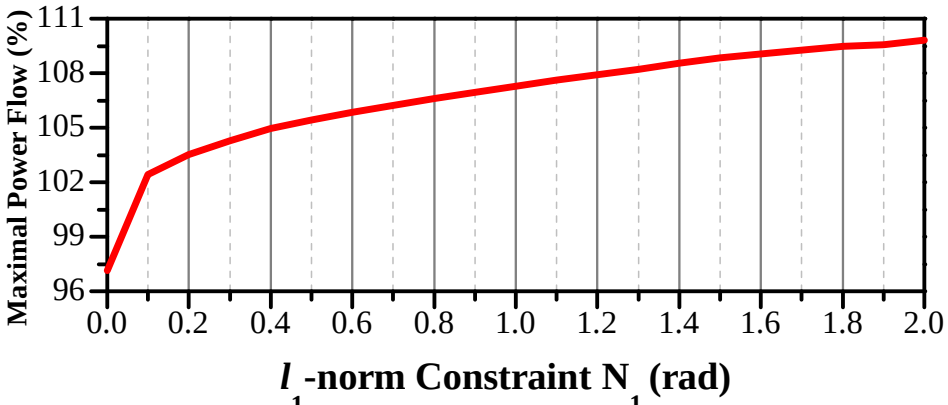

Figure 3.6: The Maximal Power Flow vs. the l_1-norm Constraint (N_1) for Target Line 2110 of the Polish System. L_S=10%.

congested before attack in each case.

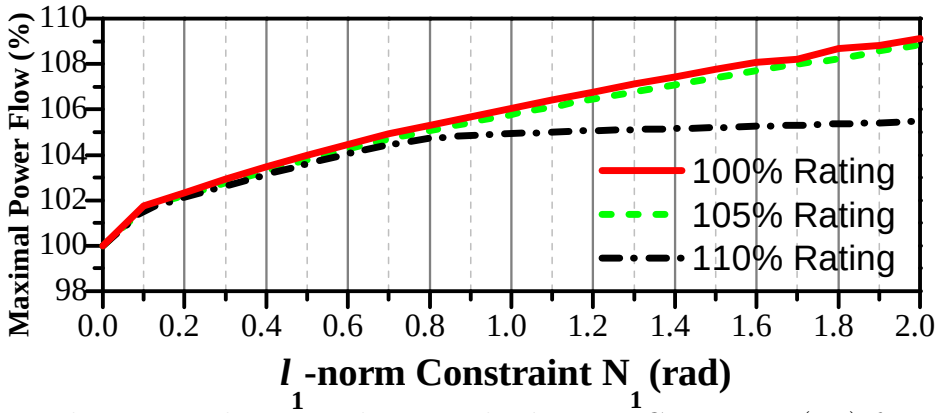

Figure 3.7: The Maximal Power Flow vs. the l_1-norm Constraint (N_1) for Target Line 292 of the Polish System under Different Congestion Levels. L_S=10%.

3.6 Conclusion

Four computationally efficient algorithms are introduced to evaluate the vulnerability of large-scale power systems to FDI attacks. Cyber-physical difference maximization (DM) can provide an upper bound of the severity of the attacks. Row generation (RG) and row and column generation (RCG) reduce the number of binary variables according to the number of critical lines and marginal generators. As the system size increases, the number of critical lines and marginal generators also in-

creases. Thus, there may still be a large number of binary variables in RG and RCG, making the attack optimization problem hard to solve in real time. However, MBD can still be applied efficiently since it only involves solving linear programs, making it more powerful in assessing large-scale system vulnerabilities. Furthermore, MBD can be easily applied to any attacker-defender bi-level linear programs. It has the flexibility to evaluate system vulnerability even with additional constraints such as ramp rate constraints, security constraints, and reserve constraints that are common in modern power system operations. Making use of these algorithms, we have also found that all critical lines are vulnerable to the proposed attacks, with the exception of radial lines with a specific configuration. Moreover, systems with higher overall congestion are more vulnerable.

Our proposed vulnerability assessment algorithms can be helpful in making the system more resilient in the following ways. Using this analysis, the system operators can identify specific lines of vulnerability, and the severity of the attacks. Certain preventive actions can be taken to prevent successful attacks. For example, if the system operators find that a line can have overflow under attack, they could artificially reduce the line limit to keep the attack from being successful. Measurements around vulnerable lines can be encrypted to prevent them from being modified. In our optimization problem, the load shift constraint characterizes the detectability of the attack, indicating that load abnormally detectors can help system operators distinguish between natural load changes and possible cyber attacks based on load redistribution.

Chapter 4

VULNERABILITY ASSESSMENT OF $N-1$ RELIABLE POWER SYSTEMS TO FDI ATTACKS

4.1 EMS Operation

In this dissertation, we consider an EMS with three core functions operating in the order of SE, RTCA, and SCED. The EMS operating structure is illustrated in Fig. 4.1. Power system measurement data collected by SCADA are sent to SE, which estimates the complex voltages after eliminating noise and bad measurements. Given the generator set points, SE also estimates the load values in the system. Modern power systems typically require $N-1$ reliability, *i.e.,* the system must operate with no violations if a contingency occurs (one of the system components, generators or branches, is out of service). RTCA simulates one power flow under each contingency k. We say a branch has a *warning* if its power flow is above a threshold η but less than its limit, while a branch has a *violation* if its power flow exceeds its limit. Note that in base case, the limit is the long-term line rating, while in contingency case it is the short-term rating. Each warning and violation generates one line limit constraint to be modeled in SCED. In contingency cases, these constraints are called security constraints. SCED takes all these constraints and solves an optimization problem to determine the most economic generation dispatch that ensures $N-1$ reliable operation.

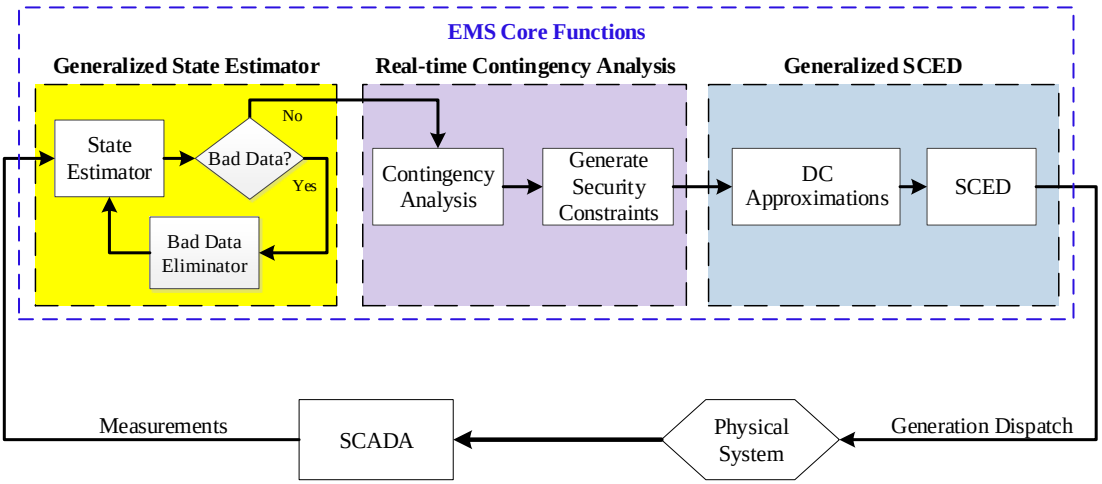

Figure 4.1: EMS Operation with SE, RTCA, and SCED.

4.2 Consequences of Attacks Designed with DCOPF on $N-1$ Reliable System

In this section, we demonstrate that attacks designed without considering RTCA and SCED (as in many existing literatures) do not cause expected physical consequences on systems operated as outlined in Fig. 4.1. The attacker's capability assumptions and the attack design ADBLP are adopted from Chapter 2. Again, the attacker is assumed to have knowledge of: (i) the complete network topology (including line parameters and ratings) and load information, and (ii) the cost, capacity, and operational status of all generators in the system. We have already seen from Chapter 3 that if the system re-dispatches using DCOPF, the attacks designed with ADBLP (2.8) can cause physical overflows.

However, modern EMSs typically operate as outlined in Fig. 4.1. Thus, the attacker cannot accurately predict the system response by solving ADBLP (2.8), and the re-dispatch after attack may not cause expected consequences. We have found in our experiments that attacks designed with DCOPF cannot cause any overflows on the synthetic Texas system operating with RTCA and SCED *even in the peak load scenario*. To illustrate this, consider the following example from our experiments.

The attacker continuously monitors the system operating status, and at the peak load hour, it observes that the most critical branch is transformer "tx-3083-3082" with a power flow of 76.72%. It selects this branch as the target and uses (2.8) (that is, modeling the system response via DCOPF) to obtain the attack vector c as well as the predicted physical power flow. It finds that the predicted flow exceeds the rating. Hence, it creates false measurements $\bar{z} = h(\hat{x} + c)$ to launch an attack. The system estimates loads from $\bar{z}$, and performs RTCA and SCED to find the optimal generation dispatch (details of RTCA and SCED are given in Sec. 4.4). Applying the new dispatch on the real loads yields the actual physical power flows. Fig. 4.2 illustrates a comparison between the attacker's predicted physical power flows and the actual flows on this target branch as a function of load shifts L_S, with $N_1 = 2$.

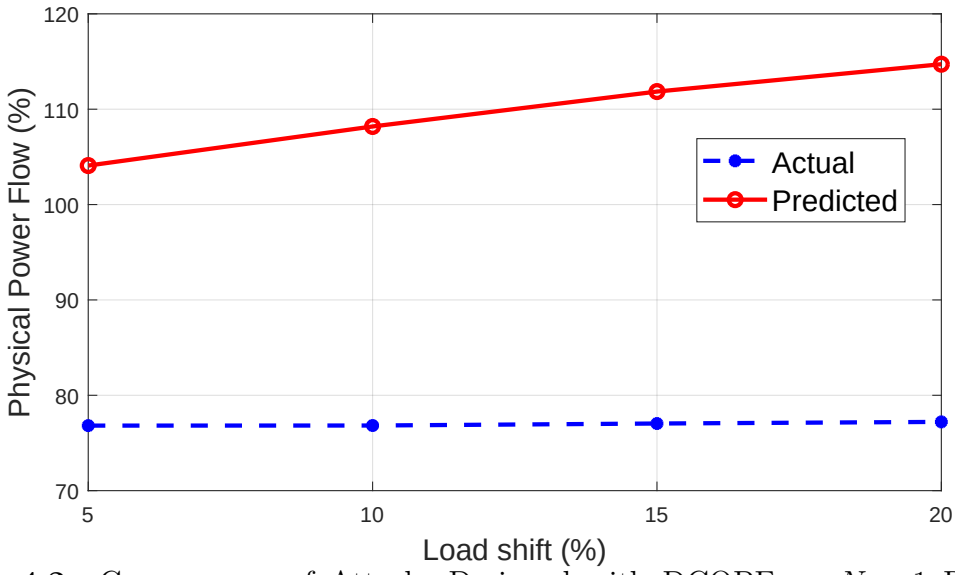

Figure 4.2: Consequence of Attacks Designed with DCOPF on $N-1$ Reliable Synthetic Texas System, $N_1 = 2$.

From this figure, we can see that the attacker predicted power flows exceed the rating of the branch for every load shift, but the actual flows are not affected. This is because in the pre-attack DCOPF solution, the target branch is congested. The attack redistributes the loads in the system, making it appear that the flow on this

branch is reduced. The higher the load shift, the more the reduction on the flow. Thus, DCOPF will re-dispatch the generations to increase the flow on this branch, making it congested again. This will overload the branch in the physical system, since the real loads are not changed. However, SCED models more constraints than DCOPF does, and this branch is congested in neither base case nor contingency cases. The load redistribution caused by the attack does not affect any binding constraints in SCED, and hence, has no effect on the re-dispatch. We have experimented on the 5 branches with highest base case flows, and observed similar consequences.

4.3 Attacker Assumptions

Among all unobservable FDI attacks, the most dangerous ones are those with serious physical consequences. In this report, we focus on a class of unobservable attacks where the attacker maliciously changes the SCADA measurements to maximize the power flow on a target line, and possibly cause overflow. The authors of [18] introduce an ADBLP to determine the worst-case unobservable line overflow attack, wherein the first level models the attacker's objective and limitations, while the second level models the system response via DCOPF. Assuming the attacker has knowledge of (i) the complete network topology (including line parameters and ratings) and load information, and (ii) the cost, capacity, and operational status of all generators in the system, the authors show that unobservable attacks found using this optimization successfully result in generation re-dispatches that cause line overflows on the IEEE RTS 24-bus system.

However, modern power systems typically do not use DCOPF to re-dispatch the generation, but rather operates as outlined in Sec. 4.1. An attacker who gains knowledge of EMS operations has an advantage to accurately predict the system response. In other words, if the attacker is able to perform the same RTCA and

SCED as the system does, it can design attacks that maximize the consequences. This is a stronger assumption than that in [18], because in addition to having access to the database of the control center, now the attacker further knows the algorithms and assumptions used by the system. While this is a stronger requirement, it is valuable to understand how the system is resilient against such strong adversaries through this worst-case approach.

In RTCA, the attacker needs to know the power flow algorithm used to get the same post-contingency flows on all lines, as well as the threshold η as described in Sec. 4.1, to determine the security constraints to be included in SCED. In SCED, the attacker should know how the system models the constraints, as different system operators may implement SCED differently. We assume the attacker has full knowledge of RTCA and SCED implementation in the EMS, in particular:

1. Contingency ratings of the branches;
2. Loss handling method;
3. Ramp rates and reserve costs of all generators;
4. Reserve policy and requirements;
5. Criteria to determine which base case line limits are to be modeled. This can be the same threshold as η in post-contingency case, but can also be different;
6. Branch flow calculation method in both base case and contingency case;
7. Load shedding policy and costs.

While it seems unrealistic to gain such knowledge, it is not entirely impossible, since such complex systems involve sophisticated (even nation-state) attackers that can exploit or have access to insider knowledge [7, 61]. Again, this is the worst-case

assumptions, and therefore, resilience of the system to such worst-case attacks can serve as an upper bound on risks to the system operations.

4.4 ADBLP to Find Worst-case Attack on $N-1$ Reliable Systems

In this section, we introduce an ADBLP similar to that in [18] to find worst-case line overflow attacks. The first level models the attacker's objective and limitations, while the second level models the system response via SCED. We focus on RTCA that simulates branch contingencies (excluding radial branches), and reports corresponding security constraints to SCED. Contingency k indicates that branch k is out of service. The attacker is assumed to be able to perform RTCA and pick a target line l to maximize its power flow when target contingency k_t occurs, and possibly create overflow. Without loss of generality, we assume the flow on l is positive; if it is not the case, its absolute value can be maximized. In the formulation below we assume the attacker aims to maximize post-contingency power flow on the target line, but the base case power flow can also be maximized. Since SCED is DC, the voltage magnitudes are all considered to be 1 p.u., and hence, c is an $n_b \times 1$ attack vector on the voltage angles.

The ADBLP takes the following form:

$$\underset{c}{\text{maximize}} \quad P_{l,k_t} - \sigma \left\|c\right\|_1 \tag{4.1a}$$

subject to

$$P_{l,k_t} = \text{OTDF}^l_{k_t}(G_B P^*_G - P_D) \tag{4.1b}$$

$$\left\|c\right\|_1 \leq N_1 \tag{4.1c}$$

$$-L_S P_D \leq Hc \leq L_S P_D \tag{4.1d}$$

$$\{P^*_G\} = \arg\left\{\min_{P_G, R_G, P, P_k} C_G\left(P_G\right) + C_R R_G\right\} \tag{4.1e}$$

subject to

$$\sum_{g=1}^{n_g} P_{Gg} = \sum_{i=1}^{n_b} P_{Di} \tag{4.1f}$$

$$\bar{P} = P_0 + \mathrm{PTDF}(G_B(P_G - P_{G0}) + Hc) \tag{4.1g}$$

$$\bar{P}_k = P_{k0} + \mathrm{OTDF}_k(G_B(P_G - P_{G0}) + Hc) \tag{4.1h}$$
$$+ \mathrm{LODF}_k \cdot \mathrm{PTDF}^k \cdot Hc, \forall k$$

$$-P_{\max} \leq \bar{P} \leq P_{\max} \tag{4.1i}$$

$$-P_{k,\max} \leq \bar{P}_k \leq P_{k,\max}, \forall k \tag{4.1j}$$

$$P_G \geq \max\{P_{G0} - M_G T_h, P_{G,\min}\} \tag{4.1k}$$

$$P_G \leq \min\{P_{G0} + M_G T_h, P_{G,\max}\} \tag{4.1l}$$

$$0 \leq R_G \leq M_G T_r \tag{4.1m}$$

$$P_G + R_G \leq P_{G,\max} \tag{4.1n}$$

$$\sum_{g=1}^{n_g} R_{Gg} \geq P_{Gg} + R_{Gg} \ , \forall g \tag{4.1o}$$

where the variables are:

c attack vector, $n_b \times 1$;

$\bar{P}, \bar{P}_k$ vectors of monitored line cyber power flows in base case and under contingency k, respectively;

P_{l,k_t} physical power flow on target line l under target contingency k_t;

P_G power output of generators, $n_g \times 1$;

R_G spinning reserve of the generators, $n_g \times 1$;

and the parameters are:

σ penalty of the l_1-norm of attack vector c;

G_B generators to buses connectivity matrix, $n_b \times n_g$;

OTDF_k outage transfer distribution factor matrix under contingency k;

OTDF_k^l l^{th} row of OTDF_k;

N_1 attack vector l_1-norm limit;

L_S load shift factor, in percentage;

H dependency matrix between power injection measurements and states, $n_b \times n_b$;

P_D vector of real loads, $n_b \times 1$;

C_G generation cost vector, $n_g \times 1$;

C_R reserve cost vector, $n_g \times 1$;

P_0, P_{k0} vectors of pre-SCED monitored line power flows in base case and under contingency k, respectively;

P_{G0} pre-SCED generator outputs, $n_g \times 1$;

PTDF power transfer distribution factor matrix;

PTDF^k k^{th} row of PTDF;

LODF_k line outage distribution factors of monitored lines under contingency k;

$P_{\max}$ vector of base case line limits;

$P_{k,\max}$ vector of line limits under contingency k;

$P_{G,\min}$ generation lower limits vector, $n_g \times 1$;

$P_{G,\max}$ generation upper limits vector, $n_g \times 1$;

M_G ramp rates of all generators, $n_g \times 1$;

T_h look-ahead time for one period SCED;

T_r time for spinning reserve requirement.

Expression in (4.1a) captures the attacker's objective of maximizing the power flow on line l under target contingency k_t, and the penalty factor σ is a small positive number to limit the attack size; constraint (4.1b) is the calculation of the power flow on line l under target contingency k_t; (4.1c) models the attacker's limited resources. Ideally, l_0-norm should be used to precisely capture the sparsity of c, but for tractability reasons we use the l_1-norm as a proxy. Constraint (4.1d) limits the percentage of load changes at each bus to avoid detection.

SCED (4.1e)-(4.1o) models the system response to the attack. The objective of the operator (4.1e) is to minimize the total cost, consisting of generation cost and reserve cost; constraint (4.1f) is the power balance equation; (4.1g) is the cyber power flow of the base case monitored lines. Note that this constraint is only modeled for the lines whose pre-SCED power flow is greater than the threshold η, *i.e.,* $|P_0/P_{\max}| \geq \eta$. This is under the assumption that the line flows will not change dramatically after the SCED re-dispatch, due to the ramping constraints of the generators. Similarly, (4.1h) is the cyber power flows on monitored lines under each contingency k, where $|P_{k0}/P_{k,\max}| \geq \eta$. Here we assume the base case and contingency case monitoring thresholds are the same. In the right hand side of (4.1h), the first term is the pre-SCED post-contingency flows; the second term is the change of the flows as a result of re-dispatch and false loads; the third term represents the amount of power on the monitored lines resulting from the effect of false loads on the contingency line k,

which is not considered in P_{k0}. Constraints (4.1i) and (4.1j) are the line limits in base case and contingency case, respectively. The active power limits in both base case and contingency cases, $P_{\max}$ and $P_{k,\max}$, are approximated from the MVA ratings and reactive flows on the branches by

$$P_{\max} = \sqrt{S_{\max}^2 - [\max(Q_{\text{from}}, Q_{\text{to}})]^2} \tag{4.2}$$

$$P_{k,\max} = \sqrt{S_{k,\max}^2 - [\max(Q_{k,\text{from}}, Q_{k,\text{to}})]^2} \tag{4.3}$$

where $S_{\max}$ and $S_{k,\max}$ are branch long-term and short-term ratings, respectively; Q_{from} and Q_{to} are the base case reactive branch flows at the "from" end and "to" end, respectively; $Q_{k,\text{from}}$ and $Q_{k,\text{to}}$ are those flows in contingency cases. Constraints (4.1k) and (4.1l) are the ramping limits; (4.1m) is the reserve limit; (4.1n) is the generation limit. Though the RTCA does not simulate generator contingencies, in SCED it is required that when a generator is out, the reserves of all other generators are sufficient to cover the output of the lost generator. We assume the SCED does not include a load shedding policy.

4.5 Simulation Results and Discussion

In this section, we present physical consequences through simulations of the attacks designed using the ADBLP described in Sec. 4.4. We use the synthetic Texas system with 2000 buses, 3210 branches, and 432 generators [62]. The topology of this system is illustrated in Fig. 4.3. The inputs to the ADBLP described in Sec. 4.4 are obtained from OpenPA [63], a Java-based EMS simulation platform that we developed in collaboration with our industry collaborators IncSys [64] and PowerData [65]. A screen shot of the simulation platform is shown in Fig. 4.4. Without attack, the system is operating at steady-state, which means that SCED does not change the generation dispatch between each EMS loop. In the base case power flow solu-

tion, the total losses among the system is 2% of the net load. We assume the SCED handles losses by uniformly increasing all loads by this percentage. RTCA simulates contingencies of all branches whose end bus voltages are both at least 100 kV, except radial branches. Prior to attack, RTCA reports no base case warnings nor violations, and 25 post-contingency warnings. We exhaustively design attacks targeting each of those 25 contingency case warnings and test the attack consequences. In our simulations, the short-term branch limit is assumed to be 115% of the long-term limit, *i.e.,* $S_{k,\max} = 115\% \times S_{\max}$; the warning threshold $\eta = 90\%$; MBD convergence tolerance $\epsilon = 5 \times 10^{-5}$; SCED look ahead time $T_h = 15$ minutes; spinning reserve time $T_r = 10$ minutes. The ADBLP is solved using Matlab with solver CPLEX on a 3.4 GHz PC with 32 GB RAM.

Figure 4.3: Synthetic Texas System Topology.

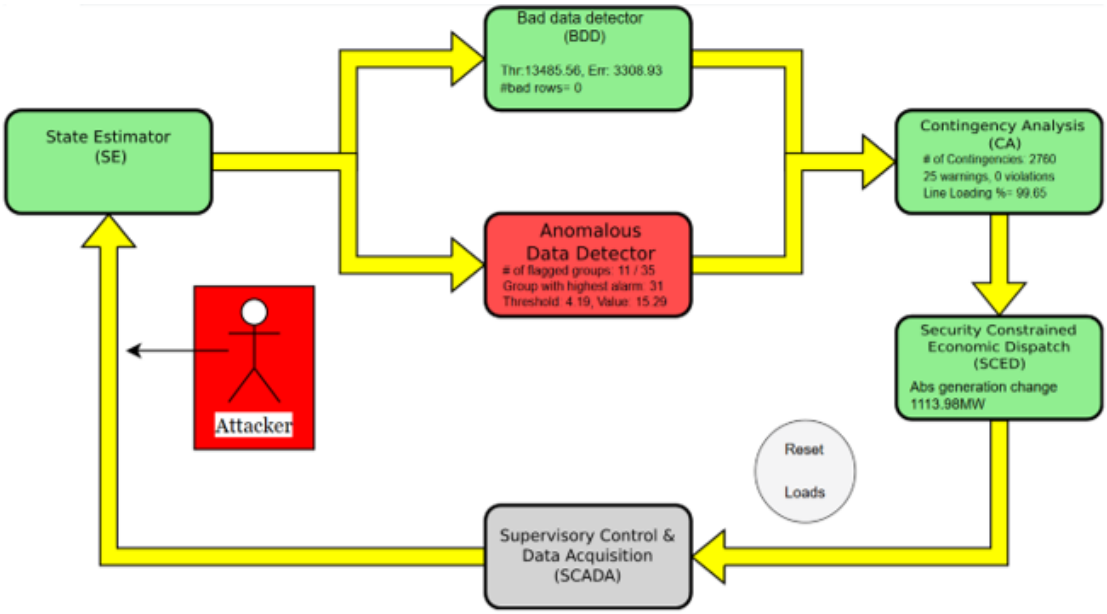

Figure 4.4: Java-based EMS Simulation Platform.

4.5.1 Approach for Attack Implementation and System Vulnerability Assessment

Fig. 4.5 illustrates the implementation of the attack and the vulnerability assessment approach. For simplicity, we assume that the real loads remain unchanged during the attack period. The physical system behavior and the SCADA measurement collection are simulated by solving an AC power flow. The true measurements z_1 from the power flow solution are acquired by the attacker to estimate the states (denoted $\hat{x}_1$). It then performs AC power flow-based RTCA to achieve the security constraints and solves the attack design ADBLP to find the attack vector c. Recall that the second level of the ADBLP is a SCED in response to the attack, and by solving it the attacker obtains an estimate on the maximal physical power flow on the target line, which is the optimal objective $\hat{P}^*_{l,k_t}$. To implement the designed attack, the attacker then constructs false measurements $\bar{z}_1 = h(\hat{x}_1 + c)$ and injects $\bar{z}_1$ to the system SE instead of the true measurements z_1. Again, only the measurements in the attack subgraph $\mathcal{S}$ are changed. Since the generator outputs are known to the system,

the false measurements will cause the SE to estimate a set of false loads. RTCA and SCED are then performed by the system to determine the new optimal generation dispatch P_G^* in response to the false loads. Once the generators re-dispatch, the attacker again acquires the true measurements z_2, and estimates the new states $\hat{x}_2$. It then sends $\bar{z}_2 = h(\hat{x}_2 + c)$ to the system SE to estimate new false loads. The system operator again runs RTCA with the new false loads and observes the cyber power flow $\bar{P}_{l,k_t}$. However, the new dispatch applied on the physical system, will maximize the physical power flow on target line l under target contingency k_t, and possibly cause overflow. The true physical power flow, P_{l,k_t}, is obtained by running RTCA with the new dispatch and real loads.

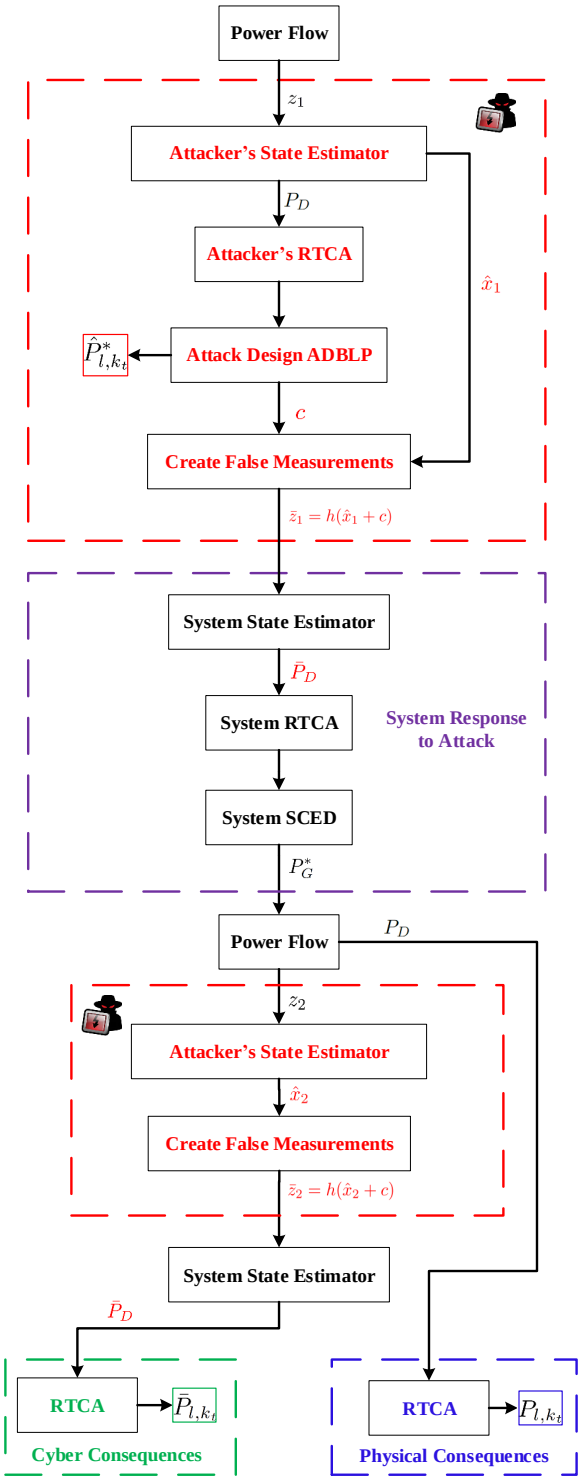

Figure 4.5: Attack Implementation and System Vulnerability Assessment Approach.

4.5.2 *Results on Maximal Physical Power Flows*

Fig. 4.6 compares physical power flow $\hat{P}^*_{l,k_t}$ predicted by the attacker, the true power flow P_{l,k_t} in the physical system, as well as the power flow (cyber) seen by the system operator $\bar{P}_{l,k_t}$, as a function of the l_1-norm constraint N_1. These power flows are plotted as percentage values relative to the active power limit $P_{l,k,\max}$ calculated using (4.3). The attacker's goal is to maximize the power flow on line 'ln-2025-2055' when line 'ln-2054-5236' is out of service.

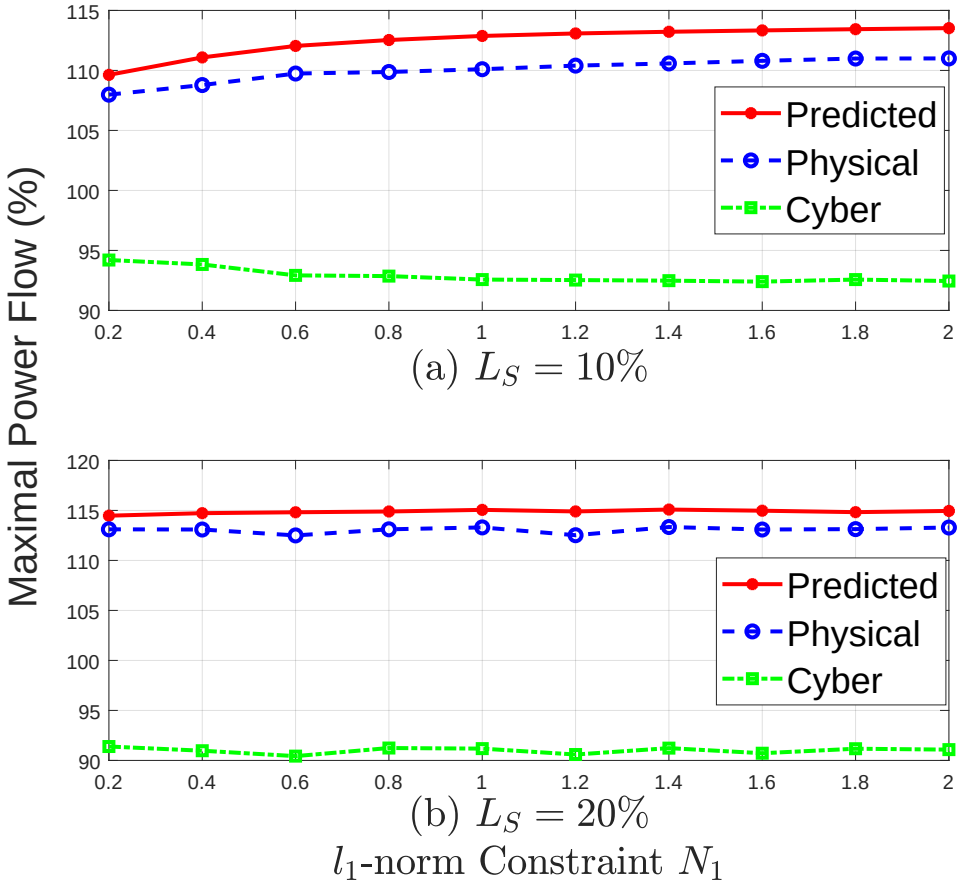

Figure 4.6: Comparison of Attacker Predicted, Physical, and Cyber Power Flows on Line 'ln-2025-2055' under Contingency 'ln-2054-5236', (a) $L_S = 10\%$; (b) $L_S = 20\%$.

When the load shift $L_S = 10\%$, $\hat{P}^*_{l,k_t}$ and P_{l,k_t} increase as N_1 increases. This indicates that the attacks are effective: they successfully cause post-contingency overflows

that cannot be seen by the system operators. When $L_S = 20\%$, similar results are observed, but $\hat{P}^*_{l,k_t}$ and P_{l,k_t} are not monotonically increasing as N_1 increases. This suggests that the MBD algorithm provides sub-optimal solutions, because as N_1 increases, the constraints are relaxed, and the optimal solution for a larger N_1 should be at least that of a smaller N_1. Maximal power flow is higher when a larger load shift is allowed. With $L_S = 20\%, N_1 = 0.2$, the power flow is higher than that when $L_S = 10\%, N_1 = 2$, which indicates that in this case load shift is the dominant constraint.

The true physical power flow P_{l,k_t} is slightly lower than the attacker predicted physical power flow $\hat{P}^*_{l,k_t}$. One possible reason for this phenomenon is that the attacker is solving a DC approximation of an AC system, and the reactive power flow may change after attack. This could result in a difference in $P_{l,k,\max}$ before and after attack. Another possible reason is that the false measurements $\bar{z}_1$ injected by the attacker cause a different set of security constraints than those that the attacker used to solve the attack design ADBLP. The attacker generates the security constraints by running RTCA using the true measurements, but those constraints generated by the system RTCA are based on the false measurements after attack. As a result, the system SCED solution may be different than the attacker predicted re-dispatch. One approach for the attacker to prevent this situation is to run its own RTCA using the false measurements and include any newly appeared security constraints into the attack design ADBLP, until there are no more new security constraints. However, this approach has no convergence guarantee, and could be too time-consuming to launch the attack in real-time.

Note that in order for the attacks to actually cause post-contingency violations requires a particular contingency to occur. Thus, the attacker has to create the target contingency itself, or gain insider knowledge about when the contingency is likely to

occur. Both are plausible for sophisticated attackers. More aggressively, the attacker can aim to create base case overflows, but the $N-1$ reliable constraints may push the system to operate conservatively. In the synthetic Texas system, there is no branch whose base case power flow is higher than η prior to the attack. Thus, to cause base case overflow, the attacker has to shift a tremendous amount of load that may easily trigger an alarm at the control center. Moreover, a large load shift will move the system operating condition dramatically with high probability, and thereby create new security constraints that are not considered when designing the attack. Thus, the consequences of the attack become unpredictable for the attacker. We have attempted to design a base case attack targeting branch 'ln-7058-7095' that has the highest base case power flow in percentage, but no overflow can be found with $L_S = 90\%$ and $N_1 = 20$. With $L_S = 100\%$ and $N_1 = 20$, the attacker's predicted power flow reaches 102.29%, but the false measurements create 3197 warnings and 24773 violations at the RTCA solution.

4.5.3 Results on Attack Resources

Fig. 4.7 illustrates the relationship between maximal power flow and l_0-norm of the attack vector (*i.e.* the number of center buses in the attack) versus the l_1-norm constraint N_1 for target line 'ln-2025-2055' under contingency 'ln-2054-5236', with different load shift constraints. As N_1 increases, so does the l_0-norm of the attack, indicating that l_1-norm is a valid proxy for l_0-norm for our problem. If a larger load shift is allowed, the maximal power flow on target line increases, but the resulting l_0-norm may decrease for the same N_1. This indicates a trade-off between load shift and attacker's resources: as the attacker attempts to avoid detection by minimizing load changes, it will require control over a larger portion of the system to launch a comparable attack.

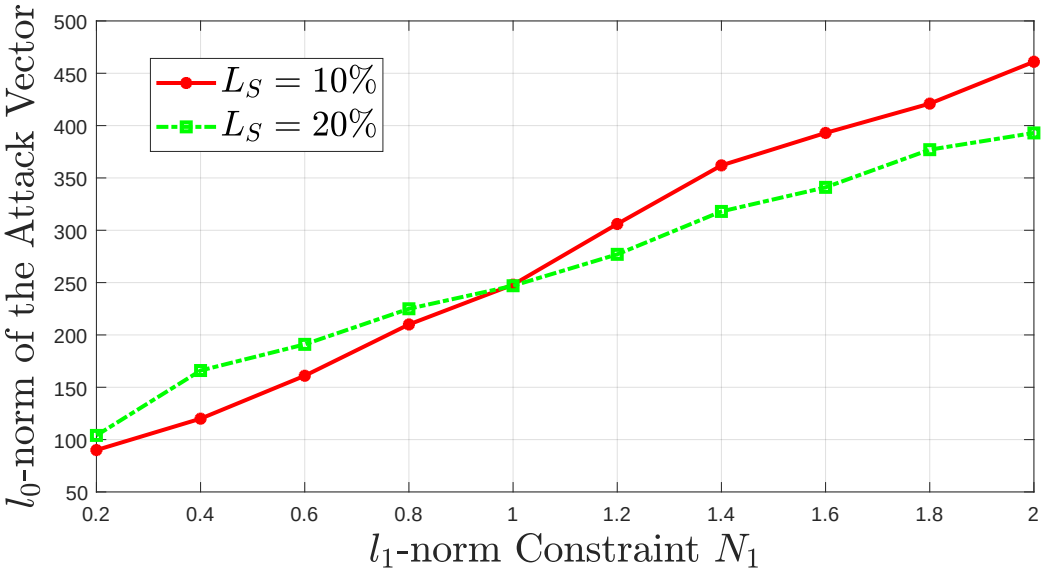

Figure 4.7: Comparison of the l_0-norm of the Attack Vector for Target Line 'ln-2025-2055' under Contingency 'ln-2054-5236'.

4.5.4 Comparison of Physical and Cyber RTCA results

Fig. 4.8 compares the physical and cyber RTCA results after the re-dispatch resulting from an attack on target line 'ln-2025-2055' under contingency 'ln-2054-5236' with load shift $L_S = 10\%, N_1 = 2$. The cyber post-contingency power flows on the x-axis represent what the system operator observes, while the y-axis represents the post-contingency power flows in the physical system. There is no point beyond 100% of the x-axis, which indicates that the system operator sees no post-contingency violation after the attack. Therefore, the attack successfully spoofed the operator that the system is in a secure state, while in reality, the target line has a 112.2% post-contingency overflow. In addition, there are four post-contingency violations that are caused by the same attack, even though they are not the attacker's targets.

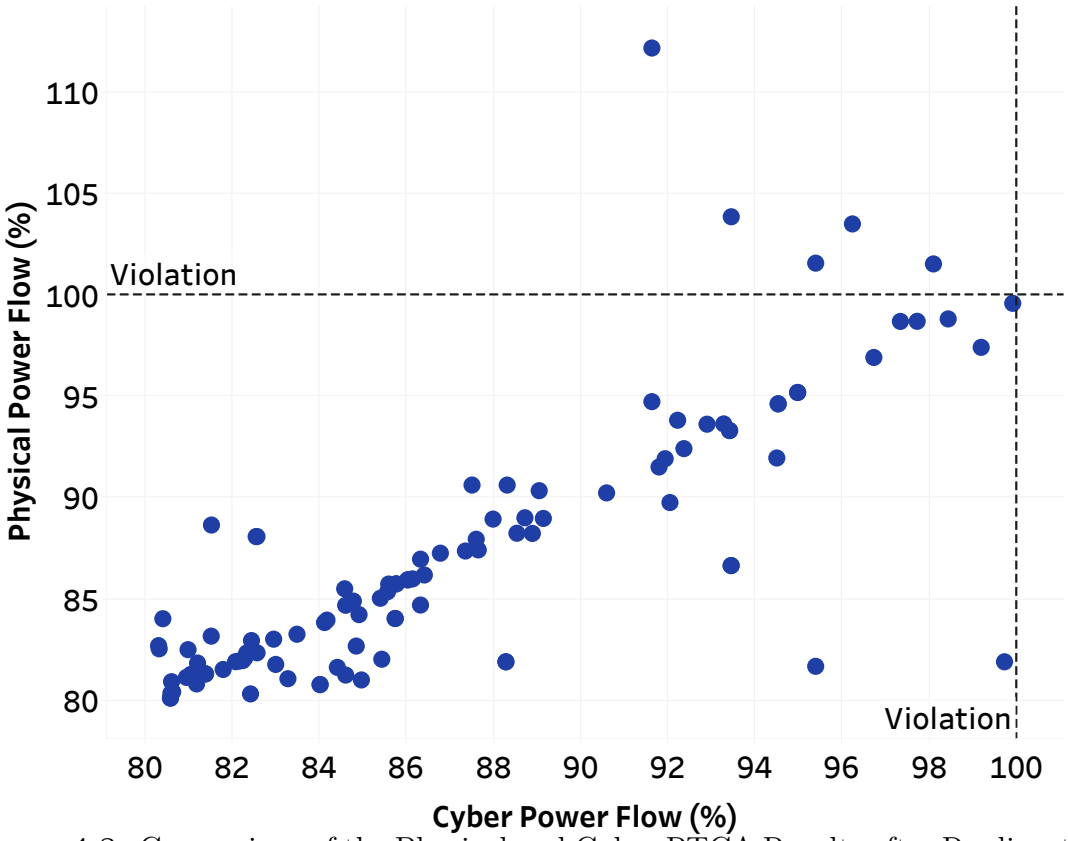

Figure 4.8: Comparison of the Physical and Cyber RTCA Results after Re-dispatch.

4.5.5 Statistical Results on Attack Consequences

As mentioned at the beginning of Sec. 3.5, we exhaustively tested attacks targeting the 25 branches with post-contingency warnings. The designed attacks successfully cause overflows on 8 out of the 25 target branches. Table 4.1 gives the statistical results on attack consequences of these 8 branches. We derived attacks using l_1-norm constraints in the range from $N_1 = 0.2$ to $N_1 = 2$. The table shows the resulting ranges in maximal power flow and l_0-norm of the attack vector c across this range. The load shift constraint $L_S = 10\%$. The prefix 'ln' indicates a transmission line and 'tx' indicates a transformer. From the maximal power flow range, we can see that some branches are more vulnerable than others, as they have higher overflows. Thus, the system operators can identify critical lines and critical contingencies for

attack protection purposes. For example, they can artificially reduce the line limit to keep the attack from being successful. Measurements around vulnerable branches can be encrypted to prevent them from being modified. In our ADBLP, the load shift constraint characterizes the detectability of the attack, indicating that load abnormally detectors can help system operators distinguish between natural load changes and possible cyber attacks based on load redistribution.

Table 4.1: Statistical Results on Maximal Physical Power Flow and l_0-norm of the Attack Vector with $N_1 \in [0.2, 2]$

Target	Contingency	Max PF (%)		$\|c\|_0$	
		N_1=0.2	N_1=2	N_1=0.2	N_1=2
ln-6188-7305	ln-7058-7095	101.92	105.08	133	442
ln-6240-6287	ln-6141-6239	102.43	106.76	137	314
ln-7233-7251	tx-6063-6062	105.41	107.90	156	485
ln-1003-1055	ln-3046-3078	102.80	102.94	163	520
ln-2025-2055	ln-2054-5236	107.98	111.00	90	461
ln-2070-5237	ln-2054-5236	101.35	104.35	90	461
ln-1003-1055	ln-1004-3133	102.43	102.56	160	513
ln-7059-7407	ln-7058-7406	100.38	102.24	154	488

4.6 Conclusion

We have evaluated the vulnerability of $N-1$ reliable power systems to unobservable FDI attacks via the physical consequences of such attacks. Such $N-1$ reliable systems are assumed to be operated by an EMS consisting of SE, RTCA, and SCED.

The attacker injects intelligently designed false measurements to the SE that bypass the bad data detector, and cause the SE to estimate false loads. The SCED re-dispatch resulting from the false loads leads to the power flow on a target line (picked by the attacker) to be maximized.

We have also highlighted the knowledge required by the attacker to design such attacks. In the worst case, the attacker can perform exactly the same RTCA and SCED as the system does, and hence, can approximately predict the system response to the attacks. Designing these attacks involves solving an ADBLP modeling the precise SCED as the system response. The designed attacks can successfully cause post-contingency overflows on target branches. Moreover, they may create more violations on branches other than the target one.

Chapter 5

UNOBSERVABLE FDI ATTACKS ON PMU MEASUREMENTS

In this chapter, we study unobservable FDI attacks against PMU measurements through their implications, and propose FIR predictive filters to detect these attacks.

5.1 PMU-based Linear State Estimation

Throughout our analysis, we assume that the power system is completely observable by PMUs. A PMU placed at a bus measures the complex voltage of that bus, and complex currents on all branches connected to it, typically at a rate of 30 samples per second [66]. These measurements are linear functions of the states, *i.e.,* the complex bus voltages. Let p be the number of buses (states), and n be the number of PMU measurements in the power system, the PMU measurement vector at each time instant, i, is given by

$$w_i = H_J x_i + e_i = \begin{bmatrix} I' \\ Y \end{bmatrix} x_i + e_i, \tag{5.1}$$

where w_i is the $n \times 1$ measurement vector; x_i is the $p \times 1$ vector of true states (complex voltages); e_i is an $n \times 1$ additive Gaussian noise vector whose covariance matrix $R = \text{diag}[\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2]$; H_J is the $n \times p$ measurement Jacobian matrix, consisting of I', the reduced identity matrix with only rows corresponding to PMU buses; and Y, the dependency matrix between available current measurements and states. The weighted least squares estimate of x_i, $\hat{x}_i$, is given by[52]

$$\hat{x}_i = (H_J^T R^{-1} H_J)^{-1} H_J^T R^{-1} w_i. \tag{5.2}$$

The conventional residue-based BDD performs chi-square test on the residue vector

$$r_{i,S} = w_i - H_J \hat{x}_i \tag{5.3}$$

to detect bad measurements. Note that the subscript S denotes state estimation; we introduce this notation in an effort to distinguish measurement residue resulting from state estimation from those resulting from using predictive algorithms that we introduce in Sec. 5.3.

5.2 FDI Attack Model on PMU Measurements

Suppose an attacker can change measurements in a set $\mathcal{S}$ by controlling a subset of PMUs. At each time instant, i, it can replace w_i with

$$\bar{w}_i = w_i + d_i, \tag{5.4}$$

where the non-zero entries of the measurement attack vector d_i are all within $\mathcal{S}$. An attack is defined to be unobservable [9] to the conventional residue-based BDD if

$$d_i = H_J c_i, \tag{5.5}$$

where the c_i is the state attack vector. Substituting (5.4) and (5.5) into (5.2) yields the estimated states $\bar{x}_i$ under attack

$$\bar{x}_i = \hat{x}_i + c_i. \tag{5.6}$$

The residue vector under attack

$$\begin{aligned} \bar{r}_{i,S} &= \bar{w}_i - H_J \bar{x}_i \\ &= w_i + d_i - H_J \hat{x}_i - H_J c_i = w_i - H_J \hat{x}_i \end{aligned} \tag{5.7}$$

is the same as that without attack. Therefore, attacks in the form of (5.5) cannot be detected by the conventional BDD.

5.3 Three Sample-based Quadratic Prediction Algorithm (TSQPA)

The residue-based BDD discussed in Sec. 5.2 does not consider temporal correlations in PMU data to detect an anomaly. To validate the quality of the incoming measurements, Gao *et al.* in [67] investigate temporal correlations in PMU data to find the relationship between the past, present, and future measurements. In particular, they prove that for loads changing at a constant power factor, the complex voltage phasor follows a quadratic trajectory. Applying auto-regressive modeling on a quadratic trajectory, they show that the vector of complex voltages at the next time instant can be predicted using the present and past states as follows:

$$x_{(i|i-1)} = 3x_{i-1} - 3x_{i-2} + x_{i-3}, \tag{5.8}$$

where $x_{(i|i-1)}$ denotes the predicted value of the complex voltage at time instant i, when the voltages at instants $i-3$ through $i-1$ are known. The authors in [67] also test the performance of TSQPA for detecting dynamic events such as the opening of transmission lines and short-circuit faults. Robustness of TSQPA for analyzing system events for different load models has been demonstrated in [68], while it was used for conditioning and validating real PMU data in [69]. However, the effectiveness of TSQPA in detecting anomalies or cyber-attacks in PMU measurements has not been investigated yet. TSQPA is emerging as a basis for real-time PMU data monitoring by some US power utilities, and therefore, it is important to evaluate its effectiveness in detecting cyber-attacks. To this end, we use TSQPA as a detector to detect anomalies due to cyber-attacks in the following way.

Applying (5.8) on estimated voltages $\hat{x}_i$ gives the predicted voltage $\hat{x}_{(i|i-1)}$. An observation residue $r_{i,T}$ (where the subscript T stands for TSQPA) at the i^{th} time instant can be obtained as:

$$r_{i,T} = \hat{x}_{(i|i-1)} - \hat{x}_i \tag{5.9}$$

If the magnitude of the observed residue $r_{i,T}$ exceeds a threshold, then a cyber-attack detection is declared.

Finally, as a point of comparison, we also consider a higher order data-driven predictive filter, for which we similarly calculate residues to detect attacks. Details of such a filter will be given in Sec. 5.6.1.

5.4 Attack Implementation

5.4.1 False Measurement Creation

We assume that the system performs DCOPF based on the measurements obtained at every five minutes [70]. After the system re-dispatches at time instant $i = 0$, the attacker solves the ADBLP (2.8) to obtain the state attack vector c, and then uses c to create false measurements. Although the loads at time instant $i = 0$ may be different than those at the fifth minute when the system re-dispatches again, it is reasonable to assume that they will not change dramatically. Hence, the attack vector solved at $i = 0$ is expected to have similar consequences to the one solved using loads at the fifth minute. Once the state attack vector c is obtained, the attacker can form a measurement attack vector d to create false measurements $\bar{w}$. However, it is unrealistic for the attacker to be omniscient and omnipotent. Thus, as mentioned in Sec. 5.2, we assume the attacker only controls a subset of PMUs, whose measurements are in $\mathcal{S}$. Given c, an attack subgraph can be constructed as in [11], consisting only of PMUs under the attacker's control. Note that here c is the outcome of the ADBLP (2.8), and hence is an attack vector on voltage angles. The measurement attack vector directly formed as $d = H_J c$ will cause loads appearing at non-load buses, and possibly raise alarm at the control center. Therefore, the attacker has to solve for the final state attack vector $\tilde{c}$ that ensures the power injections at non-load buses

remain unchanged, using the Newton-Raphson method as described in [16]. Once $\tilde{c}$ is obtained, the measurement attack vector can be constructed as $d = H_J \tilde{c}$.

5.4.2 Attack Strategies

We consider the following two strategies for the attacker to inject false measurements:

(1) Sudden attack. At any time instant on or before the fifth minute, the attacker injects d, the measurement attack vector computed at $i = 0^+$, and keeps injecting d afterwards. Without loss of generality, we focus on the situation where d is injected at the fifth minute. Denoting i as the sample number, the fifth minute is $i = 9000$ assuming PMU outputs at 30 samples/sec. The false measurements in a sudden attack are given by

$$\bar{w}_i = \begin{cases} w_i, & i < 9000 \\ w_i + d, & i \geq 9000 \end{cases}. \tag{5.10}$$

A sudden attack will cause the system to re-dispatch according to the false loads, and maximize the physical power flow on the target branch. However, as we will demonstrate in Sec. 5.6, sudden attacks can be detected by predictive filters such as TSQPA.

(2) Ramping attack. In this strategy, the attacker gradually increases the attack magnitude during the first five-minute interval, starting at $i = 1$, ensuring d is injected at the fifth minute, and keeps injecting d afterwards. The false measurements in a ramping attack are given by

$$\bar{w}_i = \begin{cases} w_i + \frac{i}{9000} \cdot d, & i < 9000 \\ w_i + d, & i \geq 9000 \end{cases}. \tag{5.11}$$

At $t = 5$ mins, the false measurements in ramping attack are identical to those in sudden attack, and hence, have the same consequences. Sec. 5.6 will illustrate that

predictive filters have more difficulty detecting ramping attacks due to the slow change across the 5 minute interval.

5.5 Generation of Synthetic Load Profile at PMU Time Scale

To verify the proposed FDI attacks against PMU-based system operations, a realistic testbed is required; specifically, the PMU measurements used to test the BDDs must reflect realistic operating conditions. In our tests, we achieve this by simulating the dynamics of the IEEE 118 bus system with time varying loads and primary generation control. The bus-level time-series load data for this test system is generated based on a real PMU dataset that was provided by a large utility company in the southwest of the US. The approach we adopted to create realistic load profiles is mainly based on the work described in [71]. The authors present a data-driven algorithm to learn from a real dataset the spatial and temporal correlation between system loads and use the learnt model to generate new synthetic data that retains the same characteristics. In [71], the approach is demonstrated on SCADA-based, hourly load data. In this section, we detail how this technique was adapted to the learning and generation of load profiles at PMU data speeds.

The utility company provided us with one week worth of PMU data for a group of neighboring substations. From the voltage and current measurements of each bus and line, we compute the loads of two substations, one at the 500kV level and one at 230kV level. Each time-series is 168 hours long, sampled at 30 samples/sec. From these two data streams we can learn the behavior of loads at different voltage levels and subsequently map them to the loads of the IEEE 118 bus system according to their voltage levels. For our simulations, we generated load data at each bus for 10 minutes. The load profile generation procedures can be found in [71], and are briefly described below.

The time-series load data for one consecutive week is broken into segments of length of 10 minutes stacked to form a load matrix P. We then factorize the load matrix P using singular value decomposition (SVD) as $P = U\Sigma V^T$. The rows of V^T, constitute the basis of the load matrix and they correspond to archetypal *temporal profiles*. The synthetic loads will be generated by taking linear combinations of a subset of the first f load basis (first f rows of V^T). Define approximations of P using first f load bases as $\hat{P} = U^f \times \Sigma^f \times V^{f^T}$, where U^f indicates the first f columns of U, Σ^f the first f columns and rows of Σ, and V^f the f first columns of V. By varying the value of f (corresponding to the number of basis vectors to be used) and measuring the root mean squared error (RMSE) between P and $\hat{P}$ we can determine an appropriate f. In Fig. 5.1, the error is plotted as a function of the number of basis vectors used. It can be seen that the error decreases rapidly up to $f = 10$ and then it slowly reaches zero when all the basis vectors are used. For this reason, the first 10 temporal profiles are used in the generation of the synthetic load profiles.

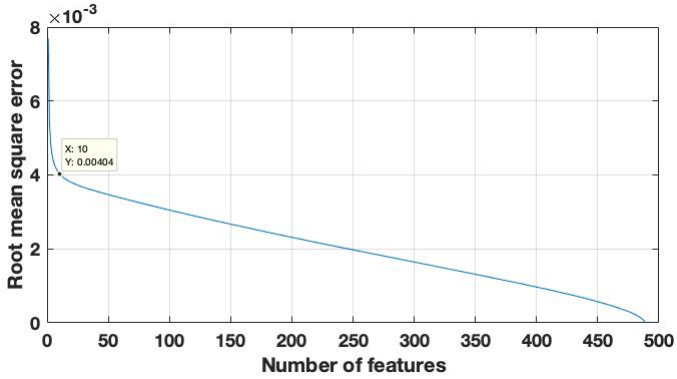

Figure 5.1: RMSE between P and $\hat{P}$ as a Function of the Number of Basis Used.

A new matrix of load profiles for n buses can then be generated as:

$$P_{\text{new}} = U^{10}_{\text{new}} \Sigma^{10} V^{10^T} \tag{5.12}$$

where P_{new}, $U^{10}_{\text{new}} \in \mathbb{R}^{n \times 10}$ is a matrix of coefficients randomly sampled from the distributions learnt from the columns of U, and Σ^{10} and V^{10^T} represent the first 10

singular values and first 10 temporal profiles obtained from the original PMU load data. To account for the spatial correlation which exists between neighboring loads, the model is modified as follows:

$$P_{\text{new}} = (DU^{10}_{\text{new}})\Sigma^{10}V^{10^T} = U'^{10}_{\text{new}}\Sigma^{10}V^{10^T} \tag{5.13}$$

where $D \in \mathbb{R}^{n\times n}$, and each entry $d_{i,j}$ of D is given by:

$$d_{i,j} = \begin{cases} 1, & \text{if } i = j \\ e^{-2\text{dist}_{i,j}}, & \text{if } \text{dist}_{i,j} \leq 3 \text{ and } i \neq j \\ 0, & \text{otherwise.} \end{cases} \tag{5.14}$$

and $\text{dist}_{i,j}$ is the minimum number of branches between buses i and j. Overall, this relation was experimentally derived in [71] and was adapted to the system for which we designed the synthetic loads.

5.6 Numerical Results

5.6.1 Experiment Setup

We use the IEEE 118-bus system in our simulations. The PMU placement scheme is obtained from [72]. The following steps are required before we can test the performance of predictive filters for attack detection:

1. Synthetic load profile generation: Using the model in (5.13) on the 500kV and 230kV loads, we generate individual load profiles for 10 minutes for the loads in the IEEE 118 bus system according to their nominal voltage. Fig. 5.2 shows the synthetic load profiles generated for two adjacent loads. As expected, they show a similar pattern over 10 minutes.

2. Synthetic PMU measurements generation: Based on the synthetic loads, dynamic simulations are run in PSLF [73] and voltage and current data are sam-

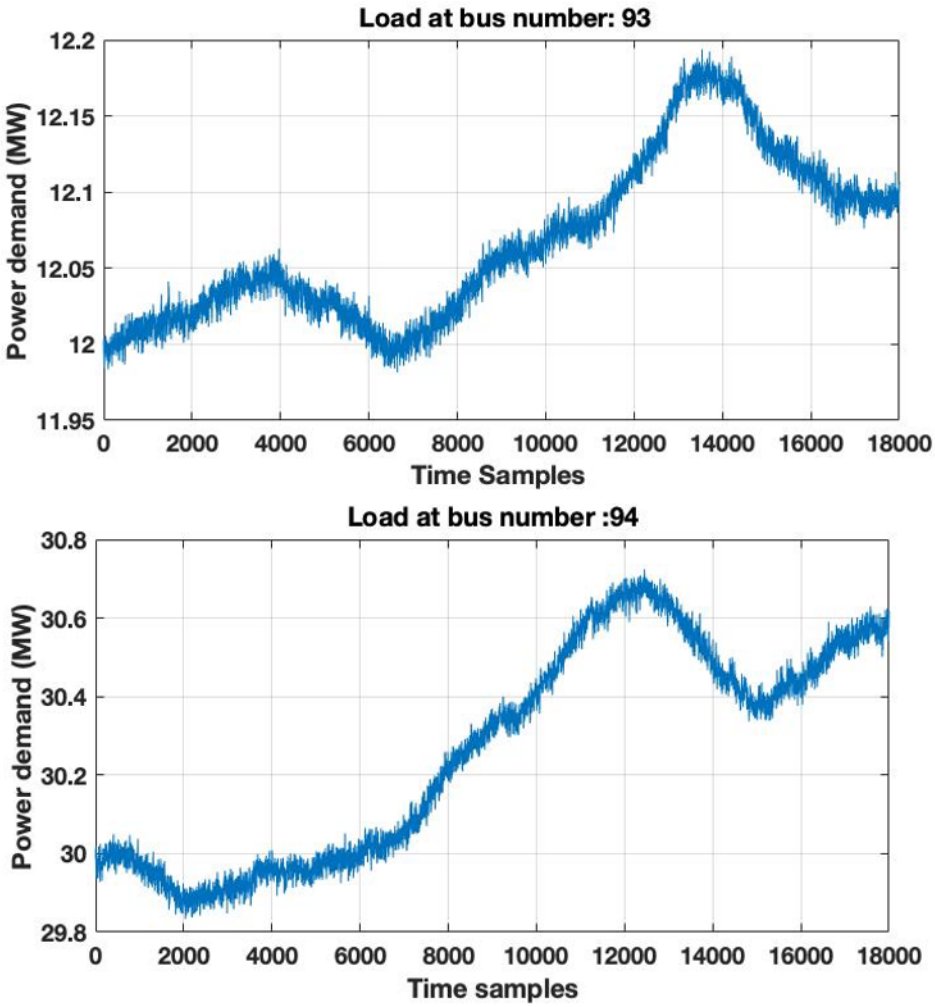

Figure 5.2: Synthetic Load Profiles Generated for Two Neighboring Buses.

pled 30 times per second to represent the PMU measurements. For adding noise to the synthetic PMU measurements we investigate the observation residues computed by TSQPA in the real PMU data obtained from the utility. The noise in the synthetic measurements are added in proportion to the noise in real data such that it results in similar observation residue for a no-attack scenario. The noise in magnitude and angle are selected from a Gaussian distribution of zero mean and 0.01% standard deviation, which ensures the total vector error (TVE) to be within 1% [74].

3. False measurements creation: A state attack vector c is obtained by solving the attack design ADBLP with 10% load shift constraint. We then follow the procedure described in Sec. 5.4 to create the measurement attack vector d, and subsequently the false measurements $\bar{w}$ for both sudden attack and ramping attack. The generation re-dispatch caused by the false measurements will lead to 30% overflow on branch 54 (bus 30-38) and 22% overflow on branch 37 (bus 8-30).

4. Data-driven five-sample predictive (FSP) filter: Based on the real PMU measurements that we received from the utility, we perform a moving window linear regression to learn the best coefficients of a five-sample predictive filter. This predictive filter is given by

$$\begin{aligned} x_{(i|i-1)} =& 0.9186x_{i-1} + 0.0196x_{i-2} + 0.0438x_{i-3} \\ &+ 0.0058x_{i-4} + 0.0122x_{i-5}. \end{aligned} \tag{5.15}$$

5.6.2 Attack Detection using Predictive Filters

We now investigate whether intelligently designed FDI attacks can be detected by predictive filters. The hypothesis of detecting an attack is that the observation residue in the presence of an attack would increase. Note that these attacks cannot be detected by the χ^2-based BDD currently employed in the power systems. False measurements resulting from sudden and ramping attack, as well as attack-free measurements at two buses of the IEEE 118 bus system are illustrated in Fig. 5.3. It can be seen that the measurements of both attack strategies are identical after 5 minutes (9,000 samples). Fig. 5.3(a) shows a relatively large attack, where the attack magnitude on the real part of the voltage at bus 8 at the fifth minute is 0.0141 per unit, while Fig. 5.3(b) shows a small attack at bus 40 where the attack magnitude to the

real part of the voltage is merely 0.0017 per unit.

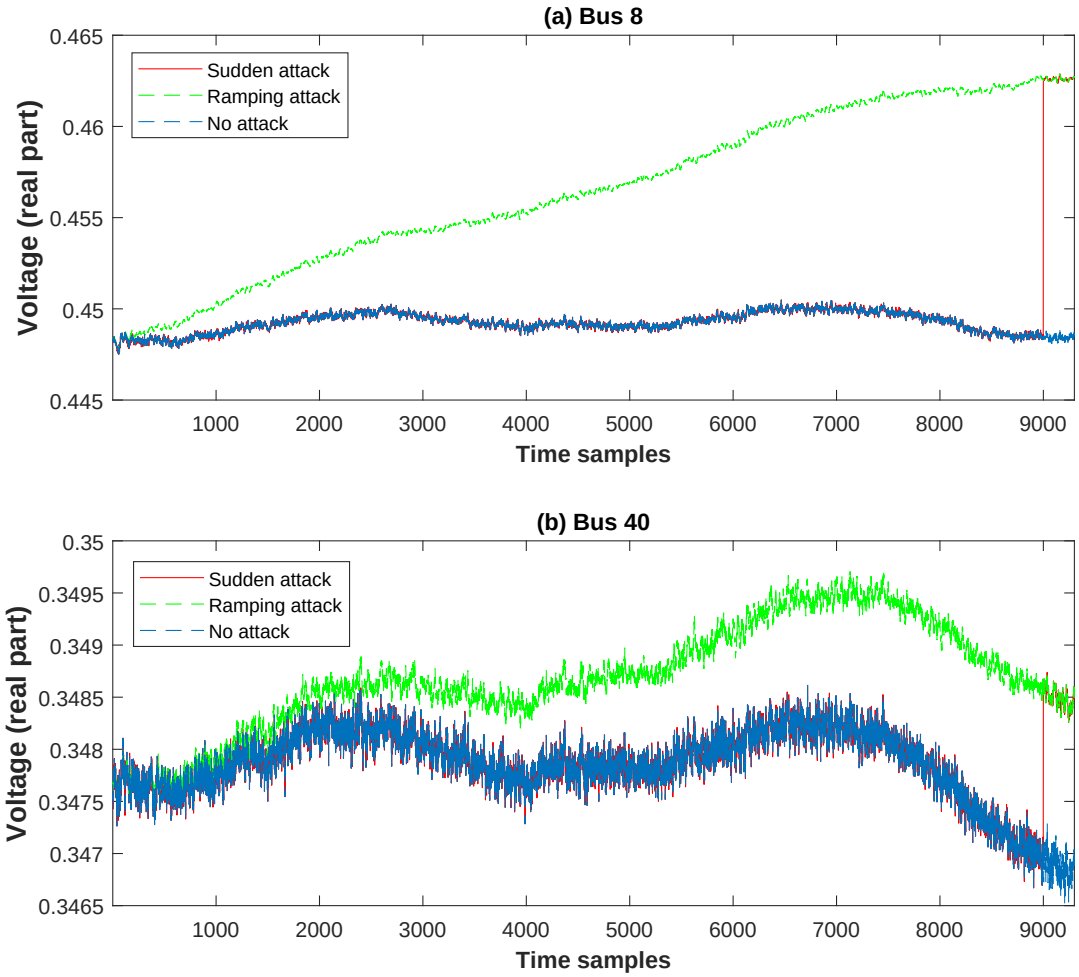

Figure 5.3: Examples of False Measurements at (a) Bus 8; and (b) Bus 40

Fig. 5.4 demonstrates the observation residues when applying the predictive filters on measurements with sudden attack. Both TSQPA and FSP give a large residue at the fifth minute when the attack is injected, indicating that they are both able to detect sudden attacks. Moreover, they can detect both the attacks at bus 8 and bus 40, even though the attack magnitude at bus 40 is much smaller.

Fig. 5.5 illustrates the observation residues obtained by applying predictive filters on measurements with ramping attack. The residues do not increase because the attack magnitude at each time instant is too small. These observations indicate that

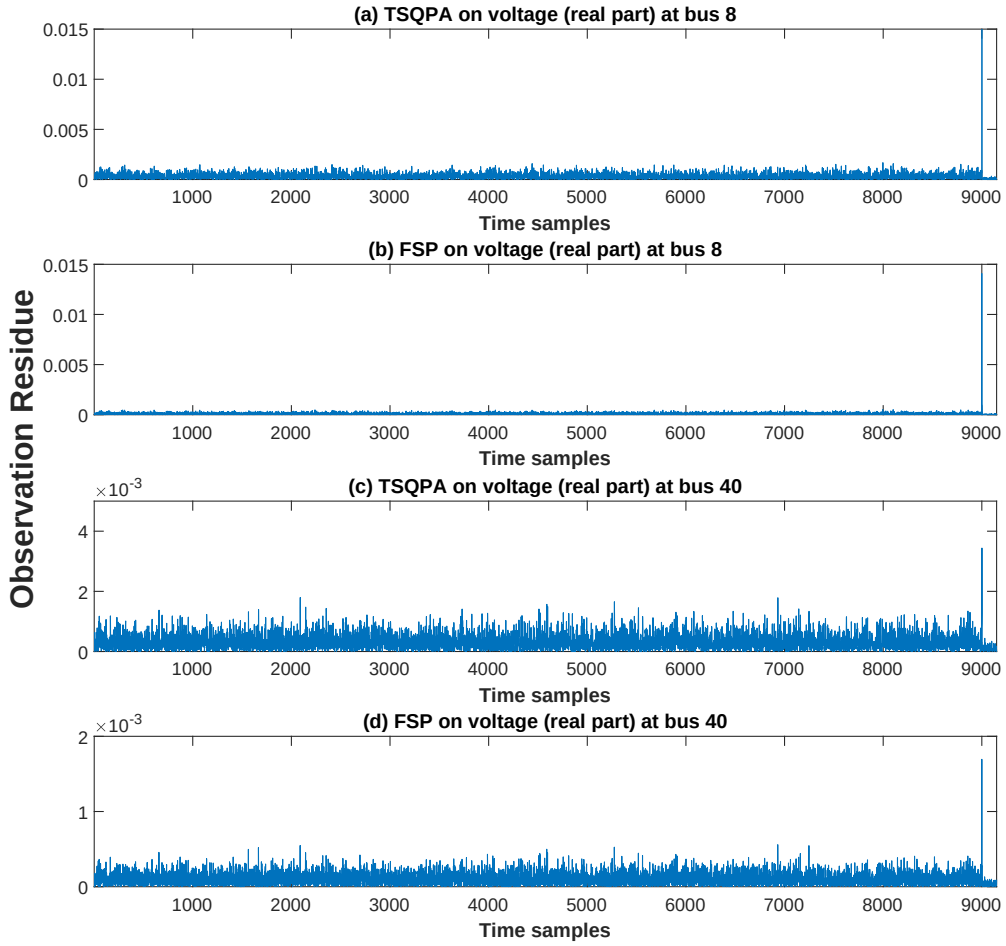

Figure 5.4: Sudden Attack Detected by Predictive Filters

gradually ramping attacks can avoid detection by the selected predictive filters.

5.7 Conclusion

In this chapter, we applied two predictive filters to detect FDI attacks against PMU measurements that are unobservable by the conventional measurement residue-based bad data detector. We first created synthetic load profiles at PMU time scale that capture both temporal and spatial correlations. Using these synthetic load profiles, we then generated synthetic PMU measurements by running dynamic simulations. Subsequently, we designed test FDI attacks via a bilevel optimization approach, and created two sets of unobservable false measurements, one for sudden attack and

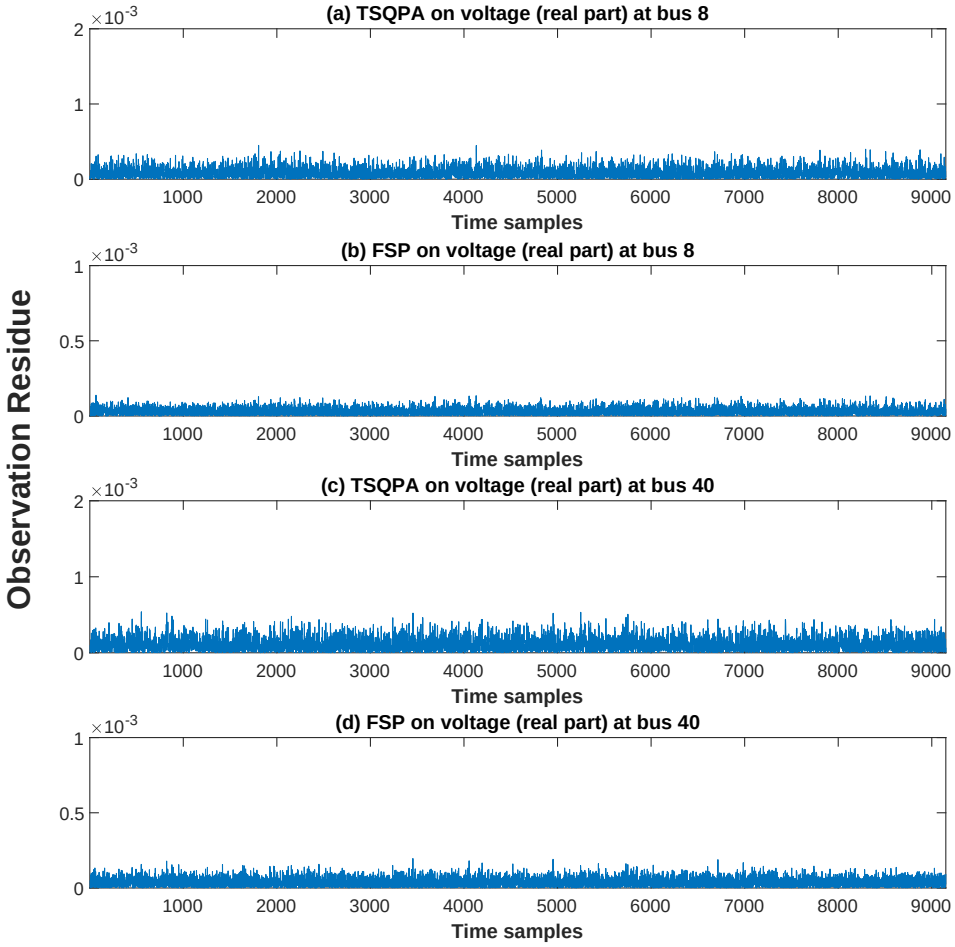

Figure 5.5: Ramping Attack Undetected by Predictive Filters

the other for ramping attack. Finally, the false measurements are tested through a theoretically derived and a data-driven predictive filter, to see whether they can detect the attacks. The observation residues obtained from the two predictive filters for both attack strategies indicate that sudden attacks can be detected by predictive filters, while ramping attacks cannot, because the ramping attack magnitudes between time instants are smaller than those of the sudden attack.

Chapter 6

UNOBSERVABLE FDI ATTACK DETECTION VIA SUPPORT VECTOR MODELS

From Chapter 3 - 5, we see that large-scale power systems, $N-1$ reliable power systems, and PMU measurements are vulnerable to FDI attacks. Therefore, it is crucial to develop countermeasures to thwart these attacks, which require the detection of the attacks as a first step. The false measurements created by the attacker results in false estimation of the loads in the system that appears as loads are re-distributed among the buses, which in turn mislead the system to incorrectly re-dispatch generation that cause consequences. Load redistribution (LR) is the reason that these attacks can affect system operations, regardless of the size of the system, functions in EMS, or which measurement system is used (SCADA/PMU). Unobservable FDI attacks are essentially LR attacks.

In this chapter, we introduce an LR attack detection framework based on support vector models by leveraging the historical load information commonly available to system operators. Unlike most existing approaches in the literature, our method determines the existence of LR attacks directly through the estimated loads, without requiring installations of new devices nor protection of specific measurements. When an LR attack occurs, the estimated loads obtained from the SE results are different from the true loads, but the net loads are the same. Thus, if accurate load predictions are available, the existence of LR attacks can be determined by comparing the predicted and estimated loads. Moreover, if an LR attack is detected, the predicted loads can be directly used to re-dispatch generation instead of using the estimated loads. By doing this, the attack consequences can be temporarily mitigated, giving

operators time to perform other corrective actions.

In particular, we propose a support vector regression (SVR) [75] based load predictor to accurately predict loads, and a subsequent support vector machine (SVM) [76] based attack detector that compares the predicted and observed loads to detect LR attacks. Our choice of this modular design aims to separate the prediction and classification, so that each module can be independently enhanced (*e.g.,* using additional features) and also replaced by other methods, as seen fit. Support vector models are optimization-based machine learning approaches that can be used for both regression and classification purposes. There are many different machine learning methods, and we choose support vector models for the following reasons: (i) they are mature methods that have been proven to be effective for various regression/classification tasks in power systems, including transient stability assessment [77], component outage estimation [78], and state estimation [79]; (ii) they are analytically developed models with fewer and easier to tune parameters compared to many other machine learning methods, *e.g.,* neural networks.

SVR has been widely used for load prediction in electric power systems. In [80], a short-term load forecasting algorithm is proposed combining SVR and particle swarm optimization. The authors of [81] proposes a SVR model that predicts very short term loads using weather data and day ahead predicted loads as features. Similar features along with additional time-related features are used to train a SVR model that predicts short term and mid term loads in [82]. In [83], Azad *et al,* predict the daily peak load using the historical peak load consumption and the corresponding temperature and relative humidity. Chong *et al,* propose a K-step ahead prediction using SVR in [84].

Proposed SVR Load Predictor: The aforementioned references focus on predicting the net load utilizing temporal correlation. To the best of our knowledge, we

are one of the first to predict loads at each bus using SVR, leveraging both spatial and temporal correlations between all the loads in the system. Features selected for the SVR predictor include historical load values of all loads chosen at distinct time intervals prior to the target time (*e.g.*, one hour before, one day before, etc.) as well as the specific time information (*e.g.,* month, weekday/weekend). This choice allows for conveniently using the same features to predict loads at different buses as the temporal features for all loads implicitly capture the spatial correlations among them.

Proposed SVM Detector: SVM is a supervised learning approach to solve classification problems, based on learning separating hyperplanes. Our approach using SVR to detect attacks largely mirrors existing approaches; our key contribution is in how we generate the training data needed to learn the SVM model to classify accurately over a large class of attacks. We now describe the dataset and our approach to train and test the two models.

Dataset: We train and test our models using the publicly available PJM metered zonal load data [85]. We map each of the 20 zones of the PJM data to a load bus in the IEEE 30-bus system, leveraging the fact that there are 20 loads in this system.

Training and Testing: To apply SVM on attack detection, it is necessary to create training data in both classes, namely *normal* and *attacked* data. The SVR predicted loads and the true loads (assuming trustworthy historical data) naturally form the normal data. For the attacked data, we propose a novel approach that generates random LR attacks in order to maximally explore the attack space, and thereby enhance accuracy in detecting *any* LR attack. Each of these attacks alters a random number of loads, and a Gaussian distribution is used to generate the deviation of each load from its true value. The severity of the attacks is controlled by varying the maximum deviation percentage over all loads. Our approach also guarantees the net

load change is 0 to satisfy the constraints of LR attacks. We use 80% of the data for training, and the remaining 20% for testing.

In addition to the random attacks, we also generate two types of intelligently designed LR attacks, namely cost maximization (CM) and line overflow (LO) attacks, to test the effectiveness of our SVM attack detector. CM attacks aim to maximize the operation cost [20]; and LO attacks attempt to overflow a target transmission line [18]. These two types of attacks are designed through optimizations to maximize their economic/physical consequences.

Our results show that the proposed attack estimation-detection framework can effectively predict and detect both random and intelligently designed LR attacks. Moreover, we illustrate that using the SVR predicted loads to re-dispatch when attacks are detected can significantly reduce the attack consequences.

The key contributions of this chapter are as follows:

1. We propose an LR attack detection framework consisting of an SVR load predictor and a subsequent SVM attack detector. This modular design enables separate enhancement of each block, and also provides sufficiently accurate predicted loads for attack mitigation purposes.

2. The SVR predictor leverages both temporal and spatial correlations within the historical load data to allow for prediction of bus-level loads. Through training and testing the proposed SVR predictor on the PJM metered load data [85], we show that it can accurately predict every load in the system.

3. Utilizing the SVR predicted loads, we train the SVM detector using normal data and random LR attacks designed to maximally explore the attack space.

4. The performance of the detection framework is tested on random attacks as well as two types of intelligently designed LR attacks. These attacks aim to cause economic/physical consequences. Our simulation results show that our detection

framework can significantly reduce the impact of LR attacks.

6.1 Load Redistribution Attacks

6.1.1 Load Redistribution (LR) Attacks and Unobservable False Data Injection (FDI) Attacks

Definition 1: LR attacks are a class of cyber-attacks that redistribute loads among the buses, while keeping the net load unchanged. The false loads in an LR attack $P_{D,\text{Atk}}$ satisfies

$$P_{D,\text{Atk}} = P_D + \Delta P_D, \tag{6.1}$$

$$\sum_i \Delta P_{Di} = 0, \tag{6.2}$$

where P_D is the true load vector, ΔP_D is the load change caused by attack, and i is the load index.

Definition 2: The load shift τ is defined to be the largest load change in percentage of the true loads:

$$\tau = \max_i \left| \frac{\Delta P_{Di}}{P_{Di}} \right| \times 100\%. \tag{6.3}$$

We use τ as an intrinsic metric to characterize the detectability of LR attacks. We found that it is trivial to detect attacks with sufficiently large τ, because load deviations far from true values are suspicious. Thus, an attacker is likely to limit τ to avoid detection by this metric. In this dissertation, we only consider LR attacks with $\tau \leq 20\%$.

The most common way to generate LR attacks is through unobservable FDI attacks against power system state estimation. Under DC power flow assumption[1], the true measurement vector z, consisting of the line power flow and bus power injection

[1]For simplicity, we focus on DC power flow settings, but our work can be generalized to AC cases as in [18].

measurements, is given by

$$z = H\theta + e, \tag{6.4}$$

where θ is the state vector (voltage angles), H is the dependency matrix between measurements and states, and e is the noise vector. Note that here we use θ instead of x to represent the states to avoid confusion in common SVR/SVM formulations, where x represents a data sample.

Recall that a false measurement vector $\bar{z}$ created with state attack vector c,

$$\bar{z} = H(\theta + c) + e, \tag{6.5}$$

is *unobservable* to the conventional bad data detector embedded with SE, because it is not distinguishable from the true measurements if the true states were $(\theta + c)$.

Let B be the dependency matrix between bus power injections and states, and let P_G be a given generation vector, then the bus power injections without attack can be expressed as

$$P_G - P_D = B\theta. \tag{6.6}$$

With attack, the false injections are given by

$$P_G - P_{D,\text{Atk}} = B(\theta + c). \tag{6.7}$$

Substituting (6.6) into (6.7) yields the load change vector

$$\Delta P_D = P_{D,\text{Atk}} - P_D = -Bc. \tag{6.8}$$

Note that since $\mathbf{1}^T B = \mathbf{0}^T$, the net load change is $\sum_i \Delta P_{Di} = -\mathbf{1}^T Bc = 0$. Thus, given a generation dispatch, an unobservable FDI attack leads to an LR attack.

6.1.2 Intelligently Designed LR Attacks

Although an attacker can inject arbitrary c as long as it controls the measurements corresponding to all non-zero entries of Hc, its goal will be to maliciously choose c

so that the resulting false loads can mislead the system re-dispatch to cause physical and/or economical consequences. We define these attacks as *intelligent attacks*, whose consequences can be maximized by solving ADBLPs. In this dissertation, we consider two specific intelligent attacks to test the robustness of our proposed detector, namely cost maximization (CM) attacks [20] and line overflow (LO) attacks [18]. The ADBLPs to find the worst case consequences of LO attacks considered here is again (2.8), and for CM attacks we only need to replace the objective function of (2.8) to be the generation cost.

As stated earlier, we want to train the SVM detector using random attacks and test its performance on intelligently designed attacks. The reason that we choose these two types of intelligent attacks is because they tend to re-distribute the loads in different directions. To illustrate this, consider the example shown in Figure. 6.1. There are two buses in this small system, and each of then has a generator and a load. We assume both the generators have generation limits far more than the amount of load in the system. One transmission line is connecting these two buses with a limit of 100 MW. The generator at bus 1 is cheaper with a cost of \$50/MWh, while generator 2 is twice as expensive. Load at bus 1 is 100MW and load at bus 2 is 200MW. The most economic dispatch is $P_{G_1} = 200\text{MW}, P_{G_2} = 100\text{MW}$, where the line is congested with power flow $P_{12} = 100\text{MW}$ and the cost is $50 \times 200 + 100 * 100 = 20\text{k\$/h}$. We assume the attacker can redistribute at most 10MW of load.

Under LO attacks, the attacker wants to maximize the power flow on the transmission line. Since the line is congested, it has to shift loads in order to make the line appear not congested, so that the system would redispatch to push more power through the line. Therefore, it needs to shift 10MW of load from bus 2 to bus 1. When the system performs load estimation, it finds that loads have changed and the line is no longer congested, so that it will redispatch to make generator 1 generate

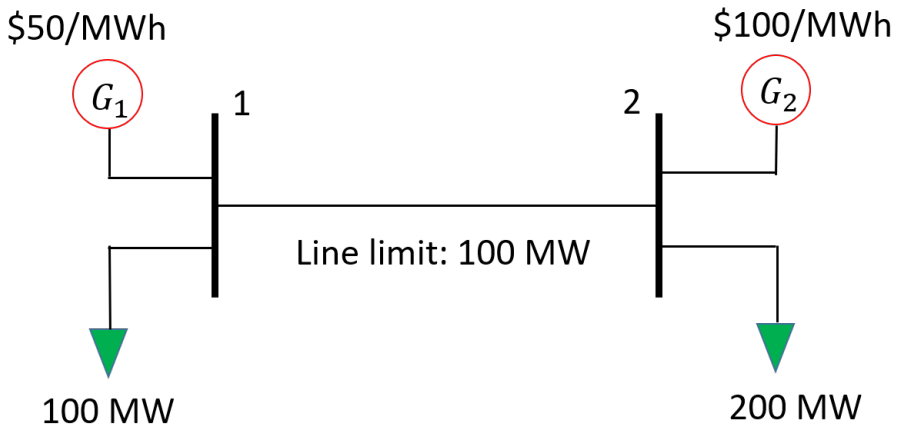

Figure 6.1: Illustrative Example Explaining Why CM and LO Attacks Tend to Redistribute Loads in Different Direction.

an extra 10MW while decrease the output of generator 2 by 10MW, because it is cheaper to do so. However the loads in the physical system never changed. As a result, the generations would be $P_{G_1} = 210\text{MW}, P_{G_2} = 90\text{MW}$, and the line flow would be $P_{12} = 110\text{MW}$, which is greater than the line limit of 100MW. However, we notice that in this case the generation cost $50 \times 210 + 100 * 90 = 19.5\text{k\$/h}$ is actually lower because the dispatch violates the line limit constraint compared to the attack-free case. This indicates that, in this case if the attacker wants to maximize the generation cost, it must redistributed reversely by moving 10MW from bus 1 to bus 2. This way, the system would see that the load has changed and there is an overflow. Thus, the system would redispatch generation to be $P_{G_1} = 190\text{MW}, P_{G_2} = 110\text{MW}$, increasing the cost to $50 \times 190 + 100 * 110 = 20.5\text{k\$/h}$. One can notice that since the true loads never changed, the power flow on the line in this case becomes $P_{12} = 90\text{MW}$ which is less than the limit. In this example, one can see that CM and LO attacks redistribute loads in completely opposite direction. In an interconnected power system with multiple congestion lines and generation limit, the CM and LO attacks may not redistribute the loads in completely opposite direction, but they will certainly move to different directions.

6.2 Proposed Attack Detection Framework

Figure 6.2 illustrates the structure of our proposed LR attack detection framework. During the real-time operation, features are selected from the historical load data until the current time step to capture both spatial and temporal correlations. Loads at the next time step are then predicted by the SVR load predictor using these features. One SVR model is trained for each load using the same features. Subsequently, the SVM attack detector takes the predicted loads and loads estimated after SE to determine the existence of LR attacks.

For detecting attacks, it should suffice to skip the SVR load predictor and plug all SVR features into the SVM to perform classification. However, in this dissertation we include the SVR for the following two reasons. The first one is that we aim to not only find an attack detection technique, but also have a corrective mechanism when attacks are detected. Using the (accurate) predicted loads to perform control actions when attacks are flagged provides time to locate the attacked measurements without causing severe consequences. The second reason is for easier scaling of the proposed models to large-scale power systems. Without the SVR predictor, the number of features used in SVM classifier will be very large, making it difficult to train and perform real-time classifications. With the SVR predictor in place, the SVM detector only needs the predicted and observed load values as features, making it useful for large-scale systems.

6.2.1 The SVR Load Predictor

Given data samples $\boldsymbol{x}_j \in \mathbb{R}^p, j = 1, 2, 3, ..., m$ and target values $\boldsymbol{y} \in \mathbb{R}^m$, an SVR attempts to find the best parameters $\boldsymbol{w}_r$ and b_r to fit $|y_j - \boldsymbol{w}_r^T \phi(\boldsymbol{x}_j) - b_r| \leq \varepsilon$ by

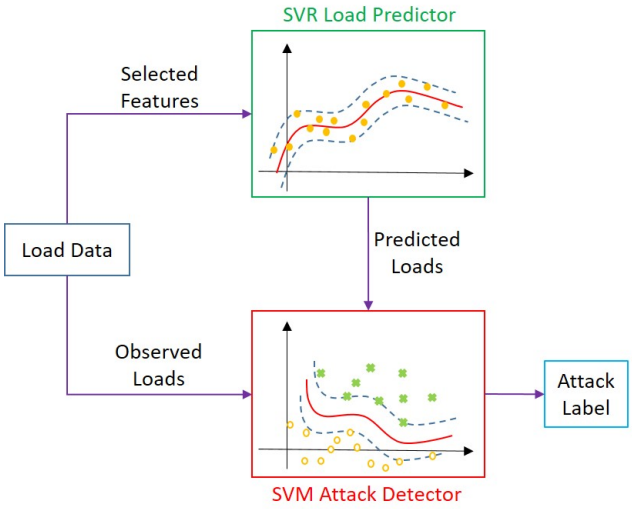

Figure 6.2: Structure of the Proposed LR Attack Detection Framework.

solving the following optimization problem [75]:

$$\underset{\boldsymbol{w}_r, b_r, \zeta_j, \zeta_j'}{\text{minimize}} \quad \frac{1}{2}\boldsymbol{w}_r^T \boldsymbol{w}_r + M \sum_{j=1}^{n} (\zeta_j + \zeta_j') \tag{6.9a}$$

$$\text{subject to} \quad y_j - \boldsymbol{w}_r^T \phi(\boldsymbol{x}_j) - b_r \leq \varepsilon + \zeta_j \quad (\alpha_j) \tag{6.9b}$$

$$\boldsymbol{w}_r^T \phi(\boldsymbol{x}_j) + b_r - y_j \leq \varepsilon + \zeta_j' \quad (\alpha_j') \tag{6.9c}$$

$$\zeta_j, \zeta_j' \geq 0, \forall j, \tag{6.9d}$$

where ε is the regression tolerance, ζ_j, ζ_j' are slack variables to allow for outliers, M is the penalty factor for outliers, α_j, α_j' are dual variables, and $\phi(\cdot)$ is a function that implicitly maps the data samples to a higher dimensional space. The dual formulation has a smaller number of variables and allows for applying the kernel trick:

$$\underset{\boldsymbol{\alpha}, \boldsymbol{\alpha'}}{\text{minimize}} \quad \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha'})^T \boldsymbol{Q}(\boldsymbol{\alpha} - \boldsymbol{\alpha'}) \tag{6.10a}$$

$$+ \varepsilon \mathbf{1}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha'}) - y^T(\boldsymbol{\alpha} - \boldsymbol{\alpha'})$$

$$\text{subject to} \quad \mathbf{1}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha'}) = 0 \tag{6.10b}$$

$$0 \leq \alpha_j, \alpha_j' \leq M, \forall j \tag{6.10c}$$

where $\boldsymbol{Q}$ is a positive semi-definite matrix, and $Q_{ij} = Q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$ is the kernel. Once the optimal solutions $(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}'^*)$ are obtained, the regression value y_{new} of a new data sample $\boldsymbol{x}_{\text{new}}$ can be computed as

$$y_{\text{new}} = \sum_{j=1}^{n} (\alpha_j^* - \alpha_j'^*) Q(\boldsymbol{x}_j, \boldsymbol{x}_{\text{new}}). \tag{6.11}$$

To accurately predict the load values, many different features can be used, including time, weather, temperature, location, and load type (residential/commercial/industrial). Intuitively, it would be the best if we use all the features to perform the prediction, but many of them are unavailable, and some of them may be redundant. The features used in the SVR load predictor also depend on the available dataset. For example, the time step of the prediction depends on how frequently the historical load data are recorded. For the specific dataset we use in this research, we select time information and historical load values at several time points relative to the target time to capture the temporal correlation, and load values at the same time points for all loads to capture the spatial correlation. Details of selected features for the SVR load predictor will be given in Section 6.3.1.

6.2.2 The SVM Attack Detector

Given data samples $\boldsymbol{u}_j \in \mathbb{R}^q, j = 1, 2, 3, ...n$ and a vector of class labels $\boldsymbol{v} \in \{1, -1\}^n$, an SVM attempts to find the decision boundary with the maximal margin to best classify $\boldsymbol{u}_j$ by solving the following optimization problem [76]:

$$\underset{\boldsymbol{w}_m, b_m, \lambda_j}{\text{minimize}} \quad \frac{1}{2} \boldsymbol{w}_m^T \boldsymbol{w}_m + C \sum_{j=1}^{n} \lambda_j \tag{6.12a}$$

$$\text{subject to} \quad v_j (\boldsymbol{w}_m^T \phi(\boldsymbol{u}_j) + b_m) \geq 1 - \lambda_j \quad (\beta_j) \tag{6.12b}$$

$$\lambda_j \geq 0, \forall j. \tag{6.12c}$$

Similar to the SVR formulation in (6.9), λ_j is a slack variable to allow for outliers, C is its penalty factor, and β_j is the dual variable. Again, applying the kernel trick, the dual formulation is used:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{Q}\boldsymbol{\beta} - \mathbf{1}^T\boldsymbol{\beta} \tag{6.13a}$$

$$\text{subject to} \quad \boldsymbol{v}^T\boldsymbol{\beta} = 0 \tag{6.13b}$$

$$0 \leq \beta_j \leq C, \forall j. \tag{6.13c}$$

Note that here $Q_{ij} = v_i v_j Q(\boldsymbol{u}_i, \boldsymbol{u}_j) = v_i v_j \phi(\boldsymbol{u}_i)^T \phi(\boldsymbol{u}_j)$. Once the optimal solution $\boldsymbol{\beta}$ is acquired, the label v_{new} for a new input data sample $\boldsymbol{u}_{\text{new}}$ can be obtained by

$$v_{\text{new}} = \text{sgn}(\sum_{j=1}^{n} v_j \beta_j^* Q(\boldsymbol{u}_j, \boldsymbol{u}_{\text{new}})) \tag{6.14}$$

where $\text{sgn}(\cdot)$ is the sign function. The features in $\boldsymbol{u}_j$ include the SVR predicted loads, the observed loads, and the same time information used in the SVR.

6.2.3 Generating Random LR Attacks to Train the SVM

We train the SVM detector using normal data and randomly designed LR attacks. The SVM detector trained using random attacks is expected to maximally explore the space of LR attacks, and hence, perform well in detecting any LR attacks. Given true loads P_D, the false loads $P_{D,\text{Atk}}$ in these random attacks are acquired using (6.1), $P_{D,\text{Atk}} = P_D + \Delta P_D$. Thus, finding $P_{D,\text{Atk}}$ is equivalent to finding ΔP_D. In each attack, we assume the attacker changes K loads at random, whose indices form a set $\mathcal{K}$, so that $\Delta P_{\mathcal{K}(k)}$ indicates the load change of the k^{th} attacked load, $k = 1, 2, \ldots, K$. The load changes of these attacked loads, denoted $\boldsymbol{\gamma}$, can be arbitrary. However, according to the LR attack property (6.2), they must be constrained to have a 0 sum. Thus, we model $\boldsymbol{\gamma}$ with a joint Gaussian distribution with 0 mean and covariance

matrix $\mathbf{\Gamma}$:

$$\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}) \tag{6.15}$$

$$\gamma_k = \Delta P_{\mathcal{K}(k)}. \tag{6.16}$$

Given a load shift τ, the diagonal entries of $\mathbf{\Gamma}$ must satisfy

$$\Gamma_{kk} = Var(\gamma_k) = (\frac{1}{2}\tau P_{\mathcal{K}(k)})^2, \forall k \tag{6.17}$$

to ensure that the probability of $|\gamma_k| \leq \tau P_{\mathcal{K}(k)}$ is 95%, because the probability of deviating beyond 2×standard deviation in a Gaussian distribution is 5%. Recall that the load changes caused by a valid LR attack must satisfy (6.2), which can be rewritten as

$$\sum_i \Delta P_{Di} = \sum_k \Delta P_{\mathcal{K}(k)} = \mathbf{1}^T \boldsymbol{\gamma} = 0. \tag{6.18}$$

Eq. (6.18) is equivalent to

$$\begin{aligned} E[(\mathbf{1}^T\boldsymbol{\gamma})^2] &= E[\mathbf{1}^T\boldsymbol{\gamma}\boldsymbol{\gamma}^T\mathbf{1}] \\ &= \mathbf{1}^T\mathbf{\Gamma}\mathbf{1} \\ &= 0. \end{aligned} \tag{6.19}$$

Finding a valid $\boldsymbol{\gamma}$ is equivalent to finding a positive semidefinite matrix $\mathbf{\Gamma}$ that satisfies (6.17) and (6.19). Since $\mathbf{\Gamma}$ is a covariance matrix, it must be positive semidefinite:

$$\mathbf{\Gamma} \succeq 0. \tag{6.20}$$

Any $\mathbf{\Gamma}$ satisfying (6.17), (6.19) and (6.20) would suffice for our application. Finding $\mathbf{\Gamma}$ is equivalent to solving a semidefinite program with arbitrary objective, constrained by (6.17), (6.19) and (6.20). The procedure to acquire false loads $P_{D,\text{Atk}}$ is summarized in Alg. 5. Varying the attack hour h, load shift τ, and number of attacked loads K, we can find feasible $\mathbf{\Gamma}$ to obtain $\boldsymbol{\gamma}$ using (6.15), and subsequently create an arbitrary

number of false loads $P_{D,\text{Atk}}$ using (6.1). Note that for specific combinations of h, τ, K, and $\mathcal{K}$, sometimes no feasible $\mathbf{\Gamma}$ can be found, but we can simply re-run Alg.5 with different inputs. Since (6.15) is drawing $\boldsymbol{\gamma}$ randomly from a Gaussian distribution, the resulting real load shift τ_r of $P_{D,\text{Atk}}$ may be different than the input τ. We keep drawing $\boldsymbol{\gamma}$ until $\tau_r \leq \tau$. The false loads created are then used to generate data samples to train and test the SVM detector.

Algorithm 5 Generating random LR attack false loads

Input: h, K, τ

Output: $P_{D,\text{Atk}}$

1. Obtain the true loads P_D at hour h.
2. Randomly select K loads to attack and let $\mathcal{K}$ denote the set of indices of the attacked loads.
3. Find a $\mathbf{\Gamma}$ satisfying (6.17), (6.19) and (6.20) with $\tau, K, \mathcal{K}$, and P_D. This can be done by solving a semidefinite program with arbitrary objective, constrained by (6.17), (6.19) and (6.20). If no feasible $\mathbf{\Gamma}$ can be found, terminate.
4. Draw the non-zero load changes $\boldsymbol{\gamma}$ from $\mathcal{N}(\mathbf{0}, \mathbf{\Gamma})$ and obtain false loads $P_{D,\text{Atk}}$ using (6.1).
5. Calculate the real load shift τ_r of $P_{D,\text{Atk}}$ using (6.3). If $\tau_r > \tau$, go to step 4). Otherwise, terminate.

6.3 Numerical Results

We use the publicly available PJM zonal hourly metered load data [85] from 2015 through 2018 for 20 transmission zones as the historical data to train and test our LR

attack detection framework. In order to conveniently create intelligently designed LR attacks as described in Section 6.1.2, we map each zone to a load bus in the IEEE 30-bus system, leveraging the fact that there are 20 loads in this system. The mapping relationship is adopted from [51], and all load values are multiplied by a scaling factor of 1.308×10^{-3} to obtain a system with moderate amount of congestion. Table 6.1 describes the mapping rules between load indices, PJM zones, and bus indices. The SVR and SVM models are implemented in Python using the Scikit-learn package [86]. The random, CM and LO attack creation are implemented in Matlab with solver Gurobi. All experiments are conducted on a 2.7 GHz CPU with 32 GB RAM.

Table 6.1: Mapping Rules between Load Indices, PJM Zones, and Bus Indices

Load	Zone	Bus	Load	Zone	Bus
1	DOM	2	11	PL	17
2	AE	3	12	PN	18
3	JC	4	13	PE	19
4	CE	7	14	RECO	20
5	AEP	8	15	ATSI	21
6	DPL	10	16	DUQ	23
7	PS	12	17	BC	24
8	DEOK	14	18	ME	26
9	PEP	15	19	EKPC	29
10	DAY	16	20	AP	30

6.3.1 The SVR Load Predictor Performance

In this section, we provide details on training and testing the SVR load predictor. As mentioned above, given the hourly load data we have, our SVR load predictor aims to accurately predict the load values at hour $h+1$ when the current hour is h. The features we use include time information and historical load values up to hour h. We select month (mo), hour (hr), and weekday/weekend (wd) as the time information features, $\boldsymbol{t} = [mo, wd, hr]$. Note that hr here is the wall clock time, for example, $hr = 14$ for 2 PM, and is different than h, which is a unique point in time. Here we only distinguish between weekdays and weekends since loads tend to be similar during weekdays, *i.e.,* $wd = 1$ for weekdays and $wd = 2$ for weekends. The temporal correlation of each load is captured by including its historical values, at hour h and s previous hours; and at hour hr and $hr+1$ of d previous days, as features. For load i, the load value features $\boldsymbol{f}_i$ are given by

$$\begin{aligned}\boldsymbol{f}_i = [&P_i^h, P_i^{h-1}, ..., P_i^{h-s}, P_i^{h-24d},\\ &P_i^{h-24d+1}, ..., P_i^{h-24}, P_i^{h-23}].\end{aligned} \tag{6.21}$$

To capture the spatial correlations, we concatenate the load value features of all the loads.

The multi-output SVR load predictor is achieved by solving one SVR optimization problem (6.9) for each load. In our experiments, we trained three SVR models to justify the contribution of capturing spatial correlations, as well as to see the influence of different selected features. Model 1 predicts each load using only time information $\boldsymbol{t}$ and its own load value features. A data sample used in Model 1 to predict load i is given by

$$\boldsymbol{x}_{j,i} = [\boldsymbol{t}, \boldsymbol{f}_i] \forall i. \tag{6.22}$$

Model 2 and 3 use $\boldsymbol{t}$ and $\boldsymbol{f}_i, \forall i$, as features to predict all loads. A data sample in these two models is given by

$$\boldsymbol{x}_j = [\boldsymbol{t}, \boldsymbol{f}_1, \boldsymbol{f}_2, ...\boldsymbol{f}_{n_l}], \tag{6.23}$$

where n_l is the number of loads in the system. In Model 2, $s = 3$ and $d = 2$; and in Model 3, $s = 4$ and $d = 3$. The ground truth $y_{j,i} = P_i^{h+1}$ is the true load value at hour $h + 1$ for load i. Table 6.2 presents some properties of the three tested SVR models. Comparing Models 1 and 2, we can see the influence of considering spatial correlations in addition to temporal correlations, as these two models use the same temporal features, but Model 2 additionally uses the features of all the loads to capture spatial correlations.

Table 6.2: Statistics of SVR Models

Model	s	d	m	p	Training time (h)
1	3	2	35011	11	1.927
2	3	2	35011	163	4.234
3	4	3	34987	223	33.324

The dimension of the data matrix $\boldsymbol{X}, m \times p$, and target value matrix $\boldsymbol{Y}, m \times n_l$, depend on the values of s and d. For example, for Model 2, $s = 3$ and $d = 2$, the length of $\boldsymbol{f}_i$ is given by

$$n_f = s + 1 + 2d = 8. \tag{6.24}$$

The resulting data sample length $p = 3 + 20 \times n_f = 163$. Since we use load values of previous $d = 2$ days as features, the start hour of our data is 01/03/2015, 0 AM. The end hour is 12/31/2018, 10 PM because for 12/31/2018, 11 PM, we do not have ground truth values of its next hour. In each of the four years, the hour when daylight

saving time ends has two load values with identical time stamps, and we approximate the load value at this hour by taking the average of those two values. As a result, the number of data samples for the SVR load predictor is

$$m = (365 \times 3 + 366 - d) * 24 - 1 - 4 = 35011. \tag{6.25}$$

The target values for hour h are the metered loads of the 20 zones at hour $h + 1$. Thus, for each data sample of length $p = 163$, the SVR outputs a vector of length 20 as prediction. We use the first 26253 data samples in year 2015 through 2017 to train the SVR load predictor and use the remaining 8758 data samples in 2018 to test its performance. The resulting training data matrix $\boldsymbol{X}_{\text{train}}$ is of size 26253×163, training target value matrix $\boldsymbol{Y}_{\text{train}}$ is of size 26253×20, testing data matrix $\boldsymbol{X}_{\text{test}}$ is of size 8758×163, and the testing target value matrix $\boldsymbol{Y}_{\text{test}}$ is of size 8758×20. The dimensions of these matrices for other models can be similarly determined.

For each model, the training data matrix $\boldsymbol{X}_{\text{train}}$ contains all data from 2015 - 2017, and data in 2018 are used as $\boldsymbol{X}_{\text{test}}$. Each column of $\boldsymbol{X}_{\text{train}}$ is scaled to zero mean and unit variance, and each column of $\boldsymbol{X}_{\text{test}}$ is scaled using the mean and variance of the corresponding column in $\boldsymbol{X}_{\text{train}}$. The same split and scaling are performed on $\boldsymbol{Y}$ to obtain $\boldsymbol{Y}_{\text{train}}$ and $\boldsymbol{Y}_{\text{test}}$ as well. The parameters in training the SVR models are chosen as $\varepsilon = 10^{-2}$ and $M = 100$. The radial basis function (RBF) kernel

$$Q(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\sigma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \tag{6.26}$$

is used with $\sigma = 10^{-2}$. Applying the trained SVR predictor on $\boldsymbol{X}_{\text{train}}$ and $\boldsymbol{X}_{\text{test}}$ yields the predicted loads $\hat{\boldsymbol{Y}}_{\text{train}}$ and $\hat{\boldsymbol{Y}}_{\text{test}}$, respectively.

Two metrics are used to evaluate the performance of the SVR load predictor, namely root mean square error (RMSE) and mean absolute percentage error (MAPE). RMSE measures the square root of the average squared error for each load, and hence

the unit is MW. MAPE measures on average how much the predicted loads deviate from their true values in percentage. These metrics for each load i are calculated as

$$\text{RMSE}_{\text{test},i} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\boldsymbol{Y}_{\text{test},i,j} - \hat{\boldsymbol{Y}}_{\text{test},i,j})^2} \tag{6.27}$$

$$\text{MAPE}_{\text{test},i} = \frac{1}{m}\sum_{j=1}^{m}\left|\frac{\boldsymbol{Y}_{\text{test},i,j} - \hat{\boldsymbol{Y}}_{\text{test},i,j}}{\boldsymbol{Y}_{\text{test},i,j}}\right| \tag{6.28}$$

where $\boldsymbol{Y}_{\text{test},i}$ is the i^{th} column of $\boldsymbol{Y}_{\text{test}}$. These metrics are used to evaluate the performance of the SVR load predictor on testing data.

Figure 6.3 illustrates the RMSE and MAPE for the SVR models. RMSE values largely depend on the load values itself, for example, load 5 has the largest RMSE value because it is the biggest load in the system. From Figure 6.3(b) we can see that the MAPE for most loads are around 1%, and MAPE for load 19, the most difficult load to predict, is around 2%. Comparing these quantities for Models 1 and 2, we can see that they are both smaller for Model 2. Recall that the difference between Models 1 and 2 is that Model 2 considers all prior loads, while Model 1 only includes the prior data at the load of interest. This result indicates that considering spatial correlations does improve the performance of the SVR load predictor. Comparing Models 2 and 3, it can be concluded that including too much historical data as features decreases the accuracy of the SVR load predictor. Besides, it can be seen from Table 6.2 that using too many features makes it extremely slow in training the SVR model. Thus, in the following sections, Model 2 is adopted to generate predicted loads used by the SVM attack detector.

In addition, we benchmark the performance of our SVR predictor against three commonly used regression techniques, namely least-squares (LS), ridge regression, and LASSO, in terms of RMSE and MAPE. Least-squares is pure linear regression. Ridge regression is least-squares linear regression with regularization on the l_2-norm

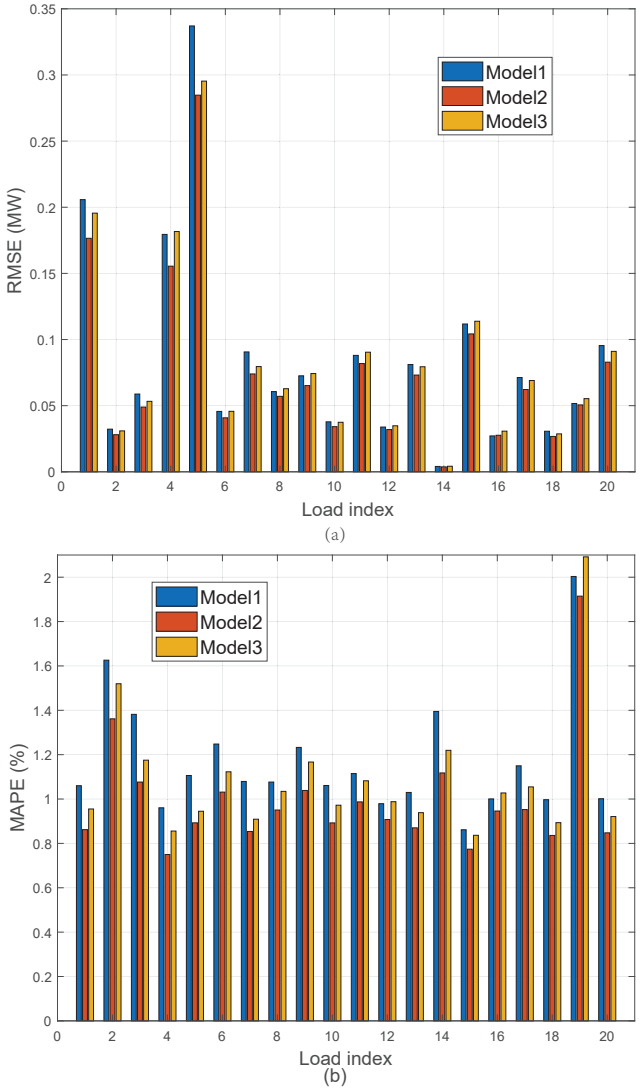

Figure 6.3: Performance of the SVR Models on Testing Data under Two Metrics: (a) RMSE, and (b) MAPE.

of the coefficients, while LASSO regularizes on the l_1-norm. Least-squares attempts to solve

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \sum_j (y_j - \boldsymbol{w}^T \boldsymbol{x}_j - b)^2. \tag{6.29}$$

With regularization, ridge regression aims to find the optimal solution to the following optimization problem

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \sum_j (y_j - \boldsymbol{w}^T \boldsymbol{x}_j - b)^2 + \rho_r \|\boldsymbol{w}\|_2^2, \tag{6.30}$$

while LASSO solves

$$\underset{\boldsymbol{w},b}{\text{minimize}} \quad \sum_j (y_j - \boldsymbol{w}^T \boldsymbol{x}_j - b)^2 + \rho_l \|\boldsymbol{w}\|_1, \tag{6.31}$$

where ρ_r and ρ_l are regularization parameters for ridge regression and LASSO, respectively.

Figures 6.4 and 6.5 illustrate the RMSE and MAPE, respectively, on testing data $\boldsymbol{X}_{\text{test}}$ of our SVR predictor (Model 2), least-squares, ridge regression, and LASSO. All models are trained in Scikit-learn using the same training data $\boldsymbol{X}_{\text{train}}$. Ridge regression and LASSO regularization parameters are $\rho_r = \rho_l = 1$. It can be seen that the SVR model outperforms the other three regression approaches, because it is capable of performing non-linear regression, while the other three can only find linear relationships.

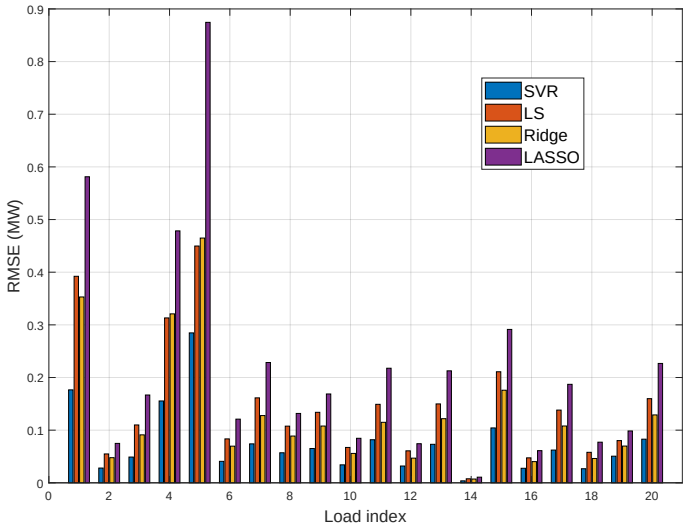

Figure 6.4: RMSE of SVR, Least-squares, Ridge Regression, and LASSO.

6.3.2 The SVM Attack Detector Performance on Random Attacks

The outputs of the SVR load predictor are used as input features of the SVM attack detector. Depending on the existence of attack, input data samples of the

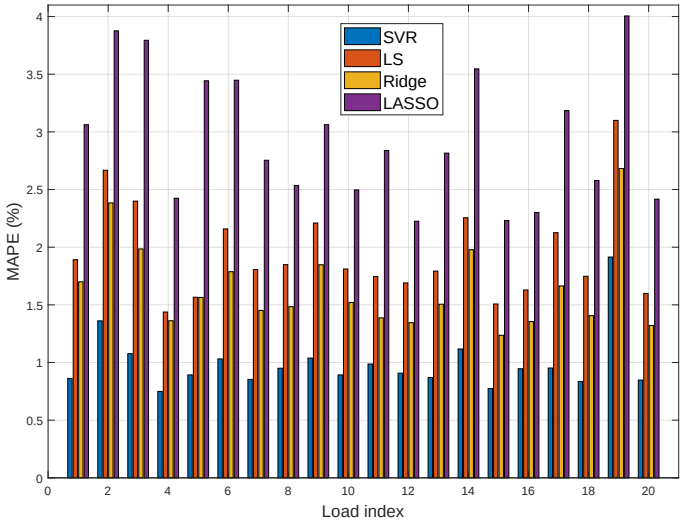

Figure 6.5: MAPE of SVR, Least-squares, Ridge Regression, and LASSO.

SVM are given by

$$\boldsymbol{u}_j = [mo, wd, hr, \hat{P}_D, P_D], \text{if } v_j = -1, \tag{6.32a}$$

$$\boldsymbol{u}_j = [mo, wd, hr, \hat{P}_D, P_{D,\text{Atk}}], \text{if } v_j = 1, \tag{6.32b}$$

where $v_j = -1$ indicates that there is no attack, and $v_j = 1$ otherwise. The predicted loads $\hat{P}_D$ of $m = 35011$ hours, along with their ground truth values P_D and time information, yield 35011 *normal* data samples for the SVM detector in the form of (6.32a). The length of each data sample $q = 3 + 20 \times 2 = 43$. The *normal* data matrix $\boldsymbol{U}_{\text{normal}}$ is of size 35011×43. We randomly select 80% of these vectors for training and the remaining 20% for testing. We create 10^5 *attacked* data samples in the form of (6.32b) using Alg. 5, resulting in $\boldsymbol{U}_{\text{attack}}$ of size $10^5 \times 43$ with real load shift τ_r ranging from 1% to 20%. From now on, we omit the subscript in τ_r for easier presentation.

We obtain different SVM models to compare their performances by varying the penalty factor C and $\tau_{\min}$ (the minimal τ used in the training data). The normal data in the training data matrix $\boldsymbol{U}_{\text{train}}$ are the same for all models, *i.e.,* the same 80% of $\boldsymbol{U}_{\text{normal}}$. The attacked data in $\boldsymbol{U}_{\text{train}}$ include 80% of attacked data samples

with $\tau \geq \tau_{\min}$. The testing data $\boldsymbol{U}_{\text{test}}$ consists of the remaining 20% of attacked data that are not used in training with all load shifts, and are the same for all models. For each model, every column of training data matrix $\boldsymbol{U}_{\text{train}}$ is scaled to zero mean and unit variance, and the same scaling is performed to the testing data. The kernel function used in the SVM detector is also the RBF kernel in the form of (6.26), but this time σ is calculated as $\sigma = 1/q$ (this is the "*scale*" option in Scikit-learn).

Figure 6.6 illustrates the effect of $\tau_{\min}$ on missed detection rate and false alarm rate. The false alarm rate is calculated by applying the detector on all $m = 35011$ normal data samples, including both training and testing. The parameter C is fixed at 1000. $\tau_{\min}$ controls the amount of attacked training data. For instance, if $\tau_{\min} = 3\%$, $\boldsymbol{U}_{\text{train}}$ contains 80% of attacks with $\tau \geq 3\%$, but does not contain any attack with $\tau < 3\%$. Intuitively, attacks with higher τ are further away from the normal data than those with lower τ. Thus, a detector trained with a low $\tau_{\min}$ will have a high false alarm rate, as the SVM is trying to find a decision boundary between normal data and attacks with small load shift. However, it should perform better in detecting attacks with small τ than detectors trained with large $\tau_{\min}$. In Figure 6.6, the blue lines indicate the missed detection rate of attacks with certain load shift τ, and the red line shows the false alarm rate. It can be seen that as $\tau_{\min}$ increases, the false alarm rate decreases, but the missed detection rate increases for attacks with small load shifts. This observation justifies the intuition discussed above, indicating that $\tau_{\min}$ is indeed a trade-off between false alarm rate and detection probability for small attacks. Note that for attacks with large τ, the effect of $\tau_{\min}$ is insignificant. For testing attacks with extremely small τ, the missed detection rates are very high even with small $\tau_{\min}$, because these attacks are in principle very difficult to detect. However, these attacks are also unlikely to cause severe consequences. From Figure 6.6, we can see that $\tau_{\min} = 3\%$ is a good choice for our dataset.

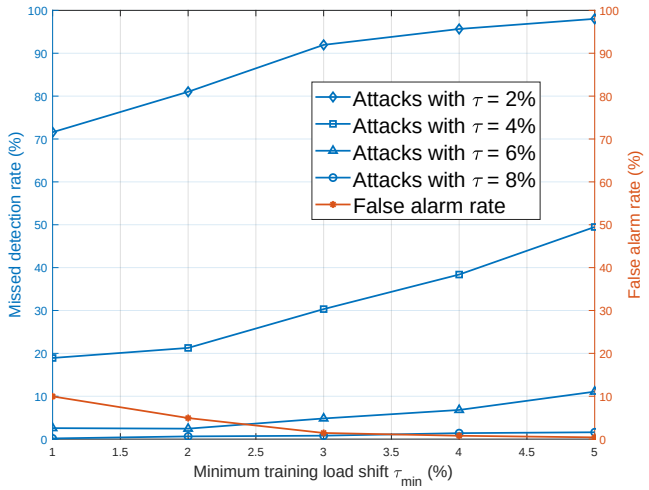

Figure 6.6: Effect of Minimum Training Load Shift $\tau_{\min}$. False Alarm Rate and Missed Detection Rate When Testing Random Attacks are Each Plotted as A Function of $\tau_{\min}$. Data is Shown for $C = 1000$.

The parameter C trades off misclassification of training examples against simplicity of the decision boundary. A small C makes the decision boundary smooth, while a large C aims at classifying all training samples correctly. Therefore, detector with large C is expected to have a better performance. However, a large C allows for fewer outliers, making it harder to solve the SVM optimization problem (6.12), so the training time increases. Figure 6.7 shows the performance of models trained with different C on testing random attacks while fixing $\tau_{\min} = 3\%$. The larger C is, the higher detection probability we can achieve. This model performs well on attacks with large τ, and the detection probability almost achieves 100% starting at $\tau = 7\%$. System operators can similarly vary $\tau_{\min}$ and C to obtain SVM model with satisfactory performance, in terms of false alarm rate and missed detection rate.

6.3.3 The SVM Attack Detector Performance on Intelligently Designed LR Attacks

In this section, we evaluate the performance of the trained SVM detector on cost maximization (CM) and line overflow (LO) attacks. According to the previous section, here we choose SVM parameters $C = 2000$ and $\tau_{\min} = 3\%$ to balance false alarm rate

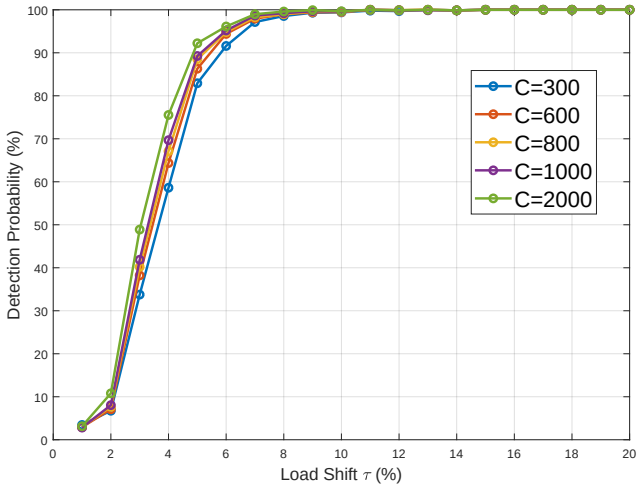

Figure 6.7: Effect of Outlier Penalty Factor C on Testing Random Attack Detection Probability. Data is Shown for $\tau_{\min} = 3\%$.

and missed detection. The procedures to generate these attacks are described as follows. On the IEEE 30-bus system, we first perform base case DCOPF for each hour in year 2015 through 2018 using the true loads. At hour h, if there are at least 2 lines whose power flows are greater than 80% of their ratings, we say those lines are *critical lines*, and h is a *critical hour*. The total number of critical hours is found to be 8038. We focus on critical hours because the false loads are likely to cause congestions at those times, which in turn change the generation dispatch to have consequences. For each critical hour, we solve optimization problem (2.8) 20 times (replacing the objective to be generation cost) to obtain attack vector c fo CM attacks with $\tau = 1\%, 2\%, \ldots, 20\%$. For each critical line, we solve (2.8) 20 times to obtain c for LO attacks, also with $\tau = 1\%, 2\%, \ldots, 20\%$. Every non-zero c is used to construct false load vector $P_{D,\text{Atk}}$ as in (6.8). If a $P_{D,\text{Atk}}$ makes the DCOPF infeasible, it is discarded. The total number of false loads for CM attacks and LO attacks are 113031 and 343135, respectively.

Figure 6.8(a) illustrates the detection probability versus the load shift τ on CM and LO attacks. For both attacks, the detection probabilities almost achieve 100%

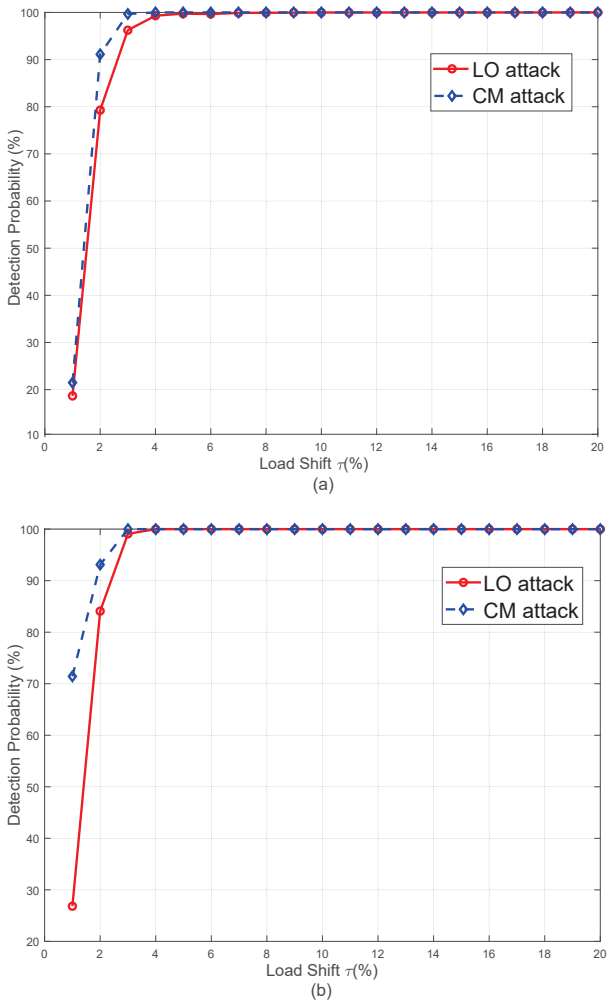

Figure 6.8: Detection Probability on CM and LO Attacks as A Function of Load Shift τ. (a) All Attacks, and (b) Attacks with Consequences. Data is Shown for $\tau_{\min} = 3\%$ and $C = 2000$.

when $\tau \geq 4\%$. For attacks with $\tau = 3\%$, the detector performance drops to 97% for LO attacks, but it is still perfect in detecting CM attacks. Comparing with the performance on random attacks as shown in Figure 6.7, it can be seen that intelligently designed attacks are easier to detect than random attacks.

Figure 6.8(b) illustrates the detection probability versus load shift τ on CM and LO attacks with consequences. CM attacks with consequences are those that increase the operating cost by more than 1%. LO attacks with consequences are those result in physical overflows. Comparing Figures 6.8(a) and 6.8(b), it can be seen that the

detector performs even better on attacks with consequences.

6.3.4 Attack Mitigation

If LR attack is flagged by our detection framework, the simplest way to mitigate the attacks is to temporarily use the loads output by the SVR load predictor for re-dispatching purposes. To test the mitigation performance using this method, we compare the worst consequences of intelligently designed attacks with and without our detection framework.

In order to obtain the consequences, we run DCOPF three times using different loads. Under normal operation, running DCOPF with true loads $P_{D,\text{normal}}$ yields the attack-free generation dispatch $P_{G,\text{normal}}$. Using attacked loads $P_{D,\text{Atk}}$ to run DCOPF gives attacked dispatch $P_{G,\text{Atk}}$. Applying $P_{G,\text{Atk}}$ on true loads $P_{D,\text{normal}}$ yields attacked line flows $P_{L,\text{Atk}} = \text{PTDF}(P_{G,\text{Atk}} - P_{D,\text{normal}})$. When an attack is detected, the system runs DCOPF using the SVR predicted loads $P_{D,\text{SVR}}$ and the resulting dispatch is $P_{G,\text{SVR}}$. The corresponding line flows are given by $P_{L,\text{SVR}} = \text{PTDF}(P_{G,\text{SVR}} - P_{D,\text{normal}})$.

Figure 6.9(a) illustrates the mitigation results for CM attacks. The word "maximum" on the y-axis indicates the worst consequence among all attacks with each load shift τ. The red line indicates the maximum cost increase without using our proposed detection framework, calculated as $a^T(P_{G,\text{Atk}} - P_{G,\text{normal}})$ (recall that a is the generation cost vector). When an attack is detected, the resulting cost increase is obtained by $a^T(P_{G,\text{SVR}} - P_{G,\text{normal}})$. When the detector fails to detect an attack, the cost increase is the attack consequence $a^T(P_{G,\text{Atk}} - P_{G,\text{normal}})$. Thus, for each load shift, if all attacks are detected, the data point on the blue line is given by $a^T(P_{G,\text{SVR}} - P_{G,\text{normal}})$. Otherwise, it is $\max[a^T(P_{G,\text{Atk}} - P_{G,\text{normal}}), a^T(P_{G,\text{SVR}} - P_{G,\text{normal}})]$. Similar procedure is performed to create Figure 6.9(b) for LO attacks. The red line is obtained by taking the maximum $P^l_{L,\text{Atk}}$ for each load shift (line l is the target line). The blue line

is obtained by $P^l_{L,\text{SVR}}$ if all attacks are detected, and $\max[P^l_{L,\text{Atk}}, P^l_{L,\text{SVR}}]$ otherwise.

Figure 6.9 illustrates the attack mitigation results for (a) CM attacks and (b) LO attacks. For each load shift, the points on the red lines indicate the worst consequence as a result of attack, and the points on the blue lines indicate the worst consequence with our attack detection framework. Points on the blue line are obtained by taking the maximum of two quantities: (i) resulting worst consequence if re-dispatch using SVR predicted loads when attack is flagged; and (ii) the worst attack consequence when the detector fails. From Figures 6.9(a), we can see that for load shift $\tau \geq 3\%$, the increases in operation cost are significantly reduced by using SVR predicted loads when an attack is flagged. For LO attacks, the overflows are significantly reduced for load shift $\tau \geq 4\%$. The largest cost increase caused by CM attacks that are not detected is 8.17% (at $\tau = 2\%$), and the largest overflow caused by LO attacks that are not detected is 3.96% (at $\tau = 3\%$). Thus, even though our detector fails to detect a small number of attacks, their consequences are minor. Note that at $\tau = 1\%$, using the SVR predicted loads leads to larger overflow due to inaccurate predictions, but the overflow is still very small. Therefore, the consequences of LR attacks can be successfully mitigated using the SVR predicted loads, which gives operators time to take other corrective actions.

6.4 Conclusion

A machine learning based load redistribution (LR) attack detection framework is proposed. This detection framework consists of a support vector regression (SVR)-based load predictor and a support vector machine (SVM)-based attack detector. The SVR load predictor is trained using features selected from historical load data to capture both spatial and temporal correlations. The prediction results indicate that the SVR load predictor can accurately predict loads at all buses. The SVM attack

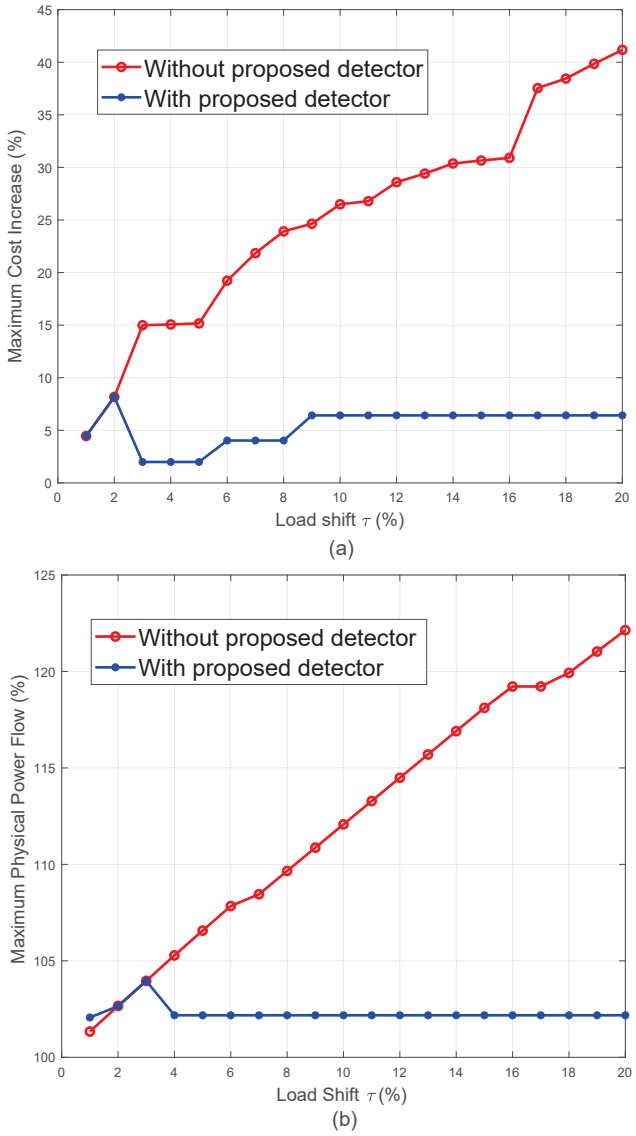

Figure 6.9: Mitigation Results of (a) CM Attacks and (b) LO Attacks.

detector is trained using randomly generated LR attacks, and is shown to be effective in detecting both randomly generated and intelligently designed attacks, especially those with consequences. Using the proposed attack detection framework, system operators can make control decisions based on the predicted loads when attack is flagged to mitigate the consequence of the attacks. It also gives operators time to find the source of the attacks. Future work will include finding attack localization techniques that help system operators identify the loads and/or meters that are modified by the attacker.

REFERENCES

[1] K. Zetter, "An unprecedented look at Stuxnet, the world's first digital weapon." https://www.wired.com/2014/11/countdown-to-zero-day-stuxnet/, Nov. 2014. 1.1

[2] B. Ibelle, "Russian cyberattack on US power grid meant to be show of power, researchers working to thwart the next one." http://news.northeastern.edu/2018/03/21/northeastern-researchers-address-russian-power-grid-attack/, Mar. 2018. 1.1

[3] FERC, "Federal Energy Regulatory Commission (FERC): Final report on the August 14th blackout in the United States and Canada: Causes and recommendations." http://www.ferc.gov/industries/electric/indus-act/reliability/blackout/ch1-3.pdf, April 2004. 1.1

[4] M. Zeller, "Myth or reality – does the Aurora vulnerability pose a risk to my generator?," in *64th Annual Conference for Protective Relay Engineers*, pp. 130–136, April 2011. 1.1

[5] S. Kelly, "Homeland security cites sharp rise in cyber attacks." http://security.blogs.cnn.com/2012/07/04/homeland-security-cites-sharp-rise-in-cyber-attacks/, July 2012. 1.1

[6] S. Toppa, "The national power grid is under almost continuous attack, report says." http://time.com/3757513/electricity-power-grid-attack-energy-security/, March 2015. 1.1

[7] K. Zetter, "Inside the cunning, unprecedented hack of Ukraine's power grid." http://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/, March 2016. 1.1, 4.3

[8] DHS, "ICS-CERT year in review reports." https://www.us-cert.gov/ics/Other-Reports. 1.1

[9] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *16th ACM Conference on Computer and Communications Security*, CCS '09, (Chicago, Illinois, USA), pp. 21–32, 2009. 1.2, 2.2, 5.2

[10] 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. 1.2

[11] 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. 1.2, 2.3, 5.4.1

[12] J. Kim and L. Tong, "On topology attack of a smart grid: Undetectable attacks and countermeasures," *IEEE JSAC*, vol. 31, no. 7, pp. 1294–1305, 2013. 1.2

[13] J. Zhang and L. Sankar, "Physical system consequences of unobservable state-and-topology cyber-physical attacks," *IEEE Transactions on Smart Grid*, vol. 7, pp. 2016–2025, July 2016. 1.2

[14] D. Kundur, X. Feng, S. Liu, T. Zourntos, and K. L. Butler-Purry, "Towards a framework for cyber attack impact analysis of the electric smart grid," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 244–249, Oct 2010. 1.2

[15] 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. 1.2

[16] J. Liang, O. Kosut, and L. Sankar, "Cyber-attacks on AC state estimation: Unobservability and physical consequences," in *IEEE PES General Meeting*, (Washington, DC), July 2014. 1.2, 5.4.1

[17] J. Zhang and L. Sankar, "Implementation of unobservable state-preserving topology attacks," in *North American Power Symposium (NAPS), 2015*, pp. 1–6, Oct 2015. 1.2

[18] J. Liang, L. Sankar, and O. Kosut, "Vulnerability analysis and consequences of false data injection attack on power system state estimation," *IEEE Transactions on Power Systems*, vol. 31, pp. 3864–3872, Sept 2016. 1.2, 1.3, 1.4, 2, 2.1, 2.3, 2.4, 2.5, 2.5, 3.4, 4.3, 4.4, 6, 1, 6.1.2

[19] L. Jia, J. Kim, R. J. Thomas, and L. Tong, "Impact of data quality on real-time locational marginal price," *IEEE Trans. Power Systems*, vol. 29, no. 2, pp. 627–636, 2014. 1.2

[20] Y. Yuan, Z. Li, and K. Ren, "Modeling load redistribution attacks in power systems," *IEEE Transactions on Smart Grid*, vol. 2, pp. 382–390, June 2011. 1.2, 6, 6.1.2

[21] L. Che, X. Liu, and Z. Li, "Mitigating false data attacks induced overloads using a corrective dispatch scheme," *IEEE Transactions on Smart Grid*, vol. 10, pp. 3081–3091, May 2019. 1.2

[22] J. Zhang, Z. Chu, L. Sankar, and O. Kosut, "Can attackers with limited information exploit historical data to mount successful false data injection attacks on power systems?," *IEEE Transactions on Power Systems*, 2018. 1.2

[23] H. Chung, W. Li, C. Yuen, W. Chung, Y. Zhang, and C. Wen, "Local cyber-physical attack for masking line outage and topology attack in smart grid," *IEEE Transactions on Smart Grid*, vol. 10, pp. 4577–4588, July 2019. 1.2

[24] Y. Xiang, L. Wang, and N. Liu, "Coordinated attacks on electric power systems in a cyber-physical environment," *Electric Power Systems Research*, vol. 149, pp. 156 – 168, 2017. 1.2

[25] J. Kang, I. Joo, and D. Choi, "False data injection attacks on contingency analysis: Attack strategies and impact assessment," *IEEE Access*, vol. 6, pp. 8841–8851, 2018. 1.2

[26] M. A. Rahman, M. H. Shahriar, M. Jafari, and R. Masum, "Novel attacks against contingency analysis in power grids." arXiv:1911.00928, 2019. 1.2

[27] Y. Yuan, Z. Li, and K. Ren, "Quantitative analysis of load redistribution attacks in power systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, pp. 1731–1738, Sept 2012. 1.2

[28] D. Alderson, G. Brown, W. Carlyle, and R. Wood, "Solving defender-attacker-defender models for infrastructure defense," in *12th INFORMS Computing Society Conference*, 2011. 1.2

[29] L. Mathiesen, "Computation of economic equilibria by a sequence of linear complementarity problems," *Mathematical Programming Study*, vol. 23, pp. 144–162, 1985. 1.2

[30] Z.-Q. Luo, J.-S. Pang, and D. Ralph, *Mathematical programs with equilibrium constraints*. Cambridge University Press, 1996. 1.2

[31] R. Andreani and J. M. Martinez, "On the solution of mathematical programming problems with equilibrium constraints," *Mathematical Methods of Operations Research*, vol. 54, no. 3, pp. 345–358, 2001. 1.2

[32] M. C. Ferris, S. P. Dirkse, and A. Meeraus, "Mathematical programs with equilibrium constraints: Automatic reformulation and solution via constrained optimization." http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.9.7017, 2002. 1.2

[33] D. W. H. Cai, S. Bose, and A. Wierman, "On the role of a market maker in networked cournot competition." http://arxiv.org/abs/1701.08896, 2017. 1.2

[34] H. Lin, Y. Deng, S. Shukla, J. Thorp, and L. Mili, "Cyber security impacts on all-PMU state estimator - a case study on co-simulation platform GECO," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, pp. 587–592, Nov 2012. 1.2

[35] C. Beasley, G. K. Venayagamoorthy, and R. Brooks, "Cyber security evaluation of synchrophasors in a power system," in *2014 Clemson University Power Systems Conference*, pp. 1–5, March 2014. 1.2

[36] J. Kim and L. Tong, "On phasor measurement unit placement against state and topology attacks," in *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 396–401, Oct 2013. 1.2

[37] H. Sedghi and E. Jonckheere, "Statistical structure learning of smart grid for detection of false data injection," in *2013 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2013. 1.2

[38] D. Lee and D. Kundur, "Cyber attack detection in PMU measurements via the expectation-maximization algorithm," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 223–227, Dec 2014. 1.2

[39] N. Dahal, R. L. King, and V. Madani, "Online dimension reduction of synchrophasor data," in *PES T&D 2012*, pp. 1–7, May 2012. 1.2

[40] Y. Chen, L. Xie, and P. R. Kumar, "Dimensionality reduction and early event detection using online synchrophasor data," in *2013 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2013. 1.2

[41] M. Wang, P. Gao, S. G. Ghiocel, J. H. Chow, B. Fardanesh, G. Stefopoulos, and M. P. Razanousky, "Identification of "unobservable" cyber data attacks on power grids," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 830–835, Nov 2014. 1.2

[42] P. Gao, M. Wang, J. H. Chow, S. G. Ghiocel, B. Fardanesh, G. Stefopoulos, and M. P. Razanousky, "Identification of successive "unobservable " cyber data attacks in power systems through matrix decomposition," *IEEE Transactions on Signal Processing*, vol. 64, pp. 5557–5570, Nov 2016. 1.2

[43] J. Zhang, Z. Chu, L. Sankar, and O. Kosut, "False data injection attacks on phasor measurements that bypass low-rank decomposition," in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2017. 1.2

[44] Z. Chu, J. Zhang, O. Kosut, and L. Sankar, "Unobservable false data injection attacks against PMUs: Feasible conditions and multiplicative attacks," in *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids*, 2018. 1.2

[45] Y. An and D. Liu, "Multivariate gaussian-based false data detection against cyber-attacks," *IEEE Access*, vol. 7, pp. 119804–119812, 2019. 1.2

[46] C. Liu, J. Wu, C. Long, and D. Kundur, "Reactance perturbation for detecting and identifying fdi attacks in power system state estimation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 763–776, Aug 2018. 1.2

[47] X. Li and K. W. Hedman, "Enhancing power system cyber-security with systematic two-stage detection strategy," *IEEE Transactions on Power Systems*, pp. 1–1, 2019. 1.2

[48] 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, pp. 1949–1959, March 2019. 1.2

[49] M. Ozay, I. Esnaola, F. T. Yarman Vural, S. R. Kulkarni, and H. V. Poor, "Machine learning methods for attack detection in the smart grid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, pp. 1773–1786, Aug 2016. 1.2

[50] D. An, Q. Yang, W. Liu, and Y. Zhang, "Defending against data integrity attacks in smart grid: A deep reinforcement learning-based approach," *IEEE Access*, vol. 7, pp. 110835–110845, 2019. 1.2

[51] A. Pinceti, L. Sankar, and O. Kosut, "Load redistribution attack detection using machine learning: A data-driven approach," in *2018 IEEE Power Energy Society General Meeting (PESGM)*, pp. 1–5, Aug 2018. 1.2, 6.3

[52] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. New York: CRC Press, 2004. 2.2, 5.1

[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004. 2, 3.4

[54] M. E. Lubbecke, "Column generation," *Wiley Encyclopedia of Operations Research and Management Science (EORMS)*, 2010. 3.1

[55] İ. Muter, Ş. İ. Birbil, and K. Bülbül, "Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows," *Mathematical Programming*, vol. 142, no. 1, pp. 47–82, 2013. 3.1

[56] ERCOT, "Electric Reliability Council of Texas (ERCOT) nodal protocols." http://ercot.com/mktrules/nprotocols/2015/02/February_2,_2015_Nodal_Protocols.pdf, January 2015. 3.1

[57] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, pp. 238–252, September 1962. 3.4

[58] A. J. Conejo, R. Minguez, E. Castillo, and R. Garcia-Bertrand, *Decomposition Techniques in Mathematical Programming*. Springer, 2006. 3.4

[59] A. M. Geoffrion, "Generalized Benders' decomposition," *Optimization Theory and Applications*, vol. 10, no. 4, 1972. 3.4

[60] N. V. Sahinidis and I. E. Grossmann, "Convergence properties of generalized Benders' decomposition," *Computers and Chemical Engineering*, vol. 15, p. 481, 1991. 3.4

[61] T.S., "The Stuxnet worm: A cyber-missile aimed at Iran," tech. rep., The Economist, 24 September 2010. 4.3

[62] TAMU, "ACTIVSg2000: 2000-bus synthetic grid on footprint of Texas." https://electricgrids.engr.tamu.edu/electric-grid-test-cases/activsg2000/, Sept. 2017. 4.5

[63] "OpenPA." https://powerdata.com/openpa/, 2020. 4.5

[64] "IncSys." http://www.incsys.com/, 2020. 4.5

[65] "PowerData." https://powerdata.com/, 2020. 4.5

[66] A. G. Phadke, J. S. Thorp, R. F. Nuqui, and M. Zhou, "Recent developments in state estimation with phasor measurements," in *IEEE/PES Power Systems Conference and Exposition*, pp. 1–7, March 2009. 5.1

[67] F. Gao, J. S. Thorp, A. Pal, and S. Gao, "Dynamic state prediction based on auto-regressive (AR) model using PMU data," in *2012 IEEE Power and Energy Conference at Illinois*, pp. 1–5, Feb 2012. 5.3, 5.3

[68] A. Pal, "Effect of different load models on the three-sample based quadratic prediction algorithm," in *2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pp. 1–5, Feb 2015. 5.3

[69] K. D. Jones, A. Pal, and J. S. Thorp, "Methodology for performing synchrophasor data conditioning and validation," *IEEE Transactions on Power Systems*, vol. 30, pp. 1121–1130, May 2015. 5.3

[70] L. Zhang, A. Bose, A. Jampala, V. Madani, and J. Giri, "Design, testing, and implementation of a linear state estimator in a real power system," *IEEE Transactions on Smart Grid*, vol. 8, pp. 1782–1789, July 2017. 5.4.1

[71] A. Pinceti, O. Kosut, and L. Sankar, "Data-Driven Generation of Synthetic Load Datasets Preserving Spatio-Temporal Features," in *IEEE Power and Energy Society General Meeting*, 2019 Accepted. 5.5, 5.5

[72] A. Pal, A. K. S. Vullikanti, and S. S. Ravi, "A PMU placement scheme considering realistic costs and modern trends in relaying," *IEEE Transactions on Power Systems*, vol. 32, pp. 552–561, Jan 2017. 5.6.1

[73] General Electric, "GE energy consulting: Positive Sequence Load Flow (PSLF)." https://www.geenergyconsulting.com/practice-area/software-products/pslf, 2020. 2

[74] IEEE, "Ieee/iec approved draft international standard - measuring relays and protection equipment - part 118-1: Synchrophasor for power system - measurements," *IEEE/IEC P60255-118-1/D8, January 2018*, pp. 1–75, Jan 2018. 2

[75] A. J. Smola and B. Schlkopf, "A tutorial on support vector regression," *Statistics and Computing*, 2004. 6, 6.2.1

[76] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, Sep 1995. 6, 6.2.2

[77] D. Yuanhang, C. Lei, Z. Weiling, and M. Yong, "Multi-support vector machine power system transient stability assessment based on relief algorithm," in *2015 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 1–5, Nov 2015. 6

[78] R. Eskandarpour and A. Khodaei, "Component outage estimation based on support vector machine," in *2017 IEEE Power Energy Society General Meeting*, pp. 1–5, July 2017. 6

[79] V. Kirincic, E. Ceperic, S. Vlahinic, and J. Lerga, "Support vector machine state estimation," *Applied Artificial Intelligence*, vol. 33, no. 6, pp. 517–530, 2019. 6

[80] S. Qiang and Y. Pu, "Short-term power load forecasting based on support vector machine and particle swarm optimization," *Journal of Algorithms & Computational Technology*, vol. 13, 2019. 6

[81] M. Capuno, J.-S. Kim, and H. Song, "Very short-term load forecasting using hybrid algebraic prediction and support vector regression," *Mathematical Problems in Engineering*, 2017. 6

[82] F. Su, Y. Xu, and X. Tang, "Short-and mid-term load forecasting using machine learning models," in *2017 China International Electrical and Energy Conference (CIEEC)*, pp. 406–411, Oct 2017. 6

[83] M. K. Azad, S. Uddin, and M. Takruri, "Support vector regression based electricity peak load forecasting," in *2018 11th International Symposium on Mechatronics and its Applications (ISMA)*, pp. 1–5, March 2018. 6

[84] L. W. Chong, D. Rengasamy, Y. W. Wong, and R. K. Rajkumar, "Load prediction using support vector regression," in *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 1069–1074, Nov 2017. 6

[85] PJM, "PJM metered hourly zonal load data," 2019. PJM Data Miner 2 https://dataminer2.pjm.com/feed/hrl_load_metered/definition. 6, 6.3

[86] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. 6.3