AN ABSTRACT OF THE THESIS OF

Carter Lassetter for the degree of Master of Science in Electrical and Computer Engineering presented on June 1, 2017.

Title: Analysis of Learning Schemes for Power Systems Secure Control

Abstract approved: _____

Eduardo Cotilla-Sanchez Jinsub Kim

As the future electrical power systems tend towards smarter and greener technology, the deployment of self-sufficient networks, or microgrids, becomes more likely. Microgrids may operate on their own or synchronized with the main grid, thus control methods need to take into account islanding and reconnecting said networks. Isolation of subnetworks may be necessary to protect either the main grid or the subnetworks themselves. With the ever growing concern of cyber-attacks on power systems, the ability to isolate network locations potentially targeted by adversaries, or exhibiting signs of cascading failures, is necessary. It is possible to create unique attacks that may leverage network operating points to maximize damage to the main grid even when the attack is confined to a microgrid. Upon isolation of a subnetwork, a control technique must be used to safely reconnect it to the main grid. The ability to optimally and safely reconnect a portion of the grid is not well understood and, as of now, limited to raw synchronization between interconnection points. A support vector machine (SVM) leveraging real-time data from phasor measurement units (PMUs) is proposed to predict in real time whether the reconnection of a sub-network to the main grid would lead to stability or instability. A dynamics simulator fed with pre-acquired system parameters is used to create training data for the SVM in various operating states. The classifier was tested on a variety of cases and operating points to ensure diversity. Accuracies of approximately 85% were observed throughout most conditions when making dynamic predictions of a given network.

Analysis of Learning Schemes for Power Systems Secure Control

by

Carter Lassetter

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 1, 2017
Commencement June 2017

Master of Science thesis of Carter Lassetter presented on June 1, 2017.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Director of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Carter Lassetter, Author

ACKNOWLEDGEMENTS

I would like to express my thanks to both of my advisers, Eduardo Cotilla-Sanchez and Jinsub Kim. Their continued guidance and expertise allowed me the opportunity to not only complete this degree, but to hone many skills needed to be prepared for my future professional career. I was lucky to have the luxury to work with both due to the different dimensions of expertise they possessed which has been a driving force to my success. My thanks to both advisers extends to my time as an Undergrad as they stimulated my interests in the field. I also appreciate the professors from EECS as a whole that have done an outstanding job teaching classes crucial to my research. I acknowledge the financial support of CREDC as well as the helpful collaboration between Universities it created. The opportunity to present and be part of the seminars every week greatly improved the efficiency and quality of my research. Id like to thank my parents Larry and Elizabeth Lassetter who have been a huge proponent towards my love of mathematics and Engineering. Finally I thank my brother and sister who have always been a source of healthy competition.

TABLE OF CONTENTS

Page

1 Introduction 1

1.1 Thesis Focus . . . 3

1.2 Thesis Layout . . . 4

2 Load Oscillating Smart Meter Attack 5

2.1 Abstract . . . 5

2.2 Introduction . . . 5

2.3 Attack Model . . . 7

2.4 Methodology . . . 9

2.5 Results . . . 11

2.6 Conclusion . . . 14

3 A Learning Scheme for Microgrid Reconnection 15

3.1 Abstract . . . 15

3.2 Introduction . . . 15

3.3 Problem Formulation and Preliminaries . . . 19

3.4 Training the SVM using a Dynamic Simulator . . . 22

3.4.1 Overview . . . 22

3.4.2 Diversifying Operating Points . . . 24

3.4.3 Dynamic Simulation . . . 24

3.4.4 Data Generation and Labeling . . . 25

3.4.5 Test Scenarios . . . 26

3.4.6 Classifier . . . 27

3.5 Results . . . 28

3.5.1 RTS-96 . . . 29

3.5.2 Poland Network . . . 32

3.6 Conclusion . . . 37

4 Security Concerns for PMU Based Controls 39

4.1 Introduction . . . 39

4.2 Raw PMU Data . . . 40

4.3 Cyber Attack . . . 40

TABLE OF CONTENTS (Continued)

Page

4.4 Attack Localizer 41
4.5 Online Data Pre-processor 42
4.6 Security Risks with No Preprocessing 44
4.7 Results with Preprocessing 45

5 Policy Based Network Control 47
5.1 Introduction 47
5.2 Dynamic Simulator 50
5.3 Policy Rollout 50
5.4 Application to Network Operation 53
5.4.1 Baseline Policies 53
5.4.2 Available Actions for Policy Rollout 54
5.5 Results 54

6 Conclusion and Future Work 57

Bibliography 58

Appendices 66
A PSS/e Models: Generator Dynamics 67

LIST OF FIGURES

Figure Page

2.1 Example microgrid attack cycling half of the load 8

3.1 Example representation of decision and error boundaries for a Support Vector Machine 20

3.2 High level overview of the process to create a classifier 23

3.3 Points of interconnection in the RTS-96 case. 28

3.4 Stable reconnection of Poland microgrid and main grid. 36

3.5 Unstable reconnection of Poland microgrid and main grid. 37

4.1 High level abstraction of detecting and localizing malicious PMU measurements 39

4.2 Training LSTM network. 42

4.3 Testing set residuals and potential thresholds. 43

5.1 Two line contingencies on the RTS-96 case. 47

5.2 Start of cascading failures in the RTS96 case. 48

5.3 Further deterioration of network operation due to cascading. 49

5.4 Policy rollout with depth one search. 51

5.5 Policy rollout with depth two search. 52

5.6 RTS-96 end load survivability with different policies. 55

5.7 RTS-96 total load survivability with different policies. 56

LIST OF TABLES

Table		Page
2.1	Different operating points for overcurrent relays	10
2.2	Undervoltage/Underfrequency load shedding relays operating points	11
2.3	Points of operation for generator protection relays	11
2.4	Remaining load, machines, branches after an attack, and amount of buses lost	12
3.1	Classifier setup after cross-validation	30
3.2	Accuracies for RTS-96 operating points independently trained	31
3.3	Accuracies for RTS-96 operating points jointly trained	32
3.4	Accuracies for RTS-96 operating points with subsets of trusted PMUs	33
3.5	Overcurrent relay configuration	33
3.6	Undervoltage/underfrequency load shedding (LS) and under/over frequency generator (GR) relay configurations	34
3.7	Baseline Poland network accuracies	34
3.8	Unseen operating point case accuracies for Poland network with subsets of trusted PMUs	35
4.1	Angular shift attack on Subnetwork Reconnection Learning Scheme	45
4.2	Analysis on 15 PMUs located near interconnection on main grid	45
4.3	Analysis on 15 PMUs located near interconnection on microgrid	46

LIST OF ALGORITHMS

Algorithm Page

LIST OF APPENDIX TABLES

Table Page

A.1 Salient generator parameters 67

A.2 Salient generator states 67

A.3 Excitation system parameters 68

A.4 Excitation system states 68

A.5 Excitation system variables 68

A.6 Governor parameters 69

A.7 Governor states 69

A.8 Governor variables 69

Chapter 1: Introduction

Electric power systems have seen great strides in recent years and will continue to improve into the future. Due to the sheer importance of energy delivery and consumption, there will always be a need to improve and maintain the electrical grid. As the world evolves around the grid, so must the grid itself.

Many changes have occurred from the beginning of electrical power systems and continue to shape the way energy is handled. Power systems have become more complex and interconnected as the system has evolved. With this increased complexity, the need for better control mechanisms arise. Electrical power systems are comprised of many components meshed together. As a result, problems in one area may end up impacting components in a nearby area. Potential mishandling or unforeseen contingencies pose problems for network operators when attempting to mitigate damages after an event occurs. Certain vulnerabilities in the grid may lay dormant until a certain contingency exposes an underlying problem creating an increased probability of cascading failures [59]. These failures are analogous to a domino effect in which one problem may expose another, toppling down until the entire network collapses.

Many techniques for guarding against blackouts exist, mainly in the form of protective relaying [63]. Load shedding is a popular method in avoiding potential voltage collapses throughout a system [11]. Load shedding takes the form of cutting power to key locations in hopes of preventing voltages from sagging to critical levels. More aggressive techniques in the form of network partitioning may take place to further defend against system collapse. In the face of cascading failures, emergency actions may take place to island the power grid into several self-sufficient networks to avoid a full scale blackout [26]. Emergency islanding was seen in the Europe blackout in 2006 in which a single overhead line trip caused the continent to divide into three main islands [37].

A popular solution that has been seeing consideration is the idea of a network made up of many smaller networks. These smaller networks are known as 'microgrids' which are normally composed of load, generation, and energy storage [38]. These microgrids may operate whilst interconnected to the main grid and have the ability to isolate from

it and continue to serve the local load within. It is clear that the ability to island from the main grid may be necessary in the face of a potential contingency on either the main grid or microgrid side in order to protect one another. With the ability to isolate oneself from a problem, a network could continue serving load while recovering from an isolated contingency. These microgrids would aid in the larger picture of a Smarter Grid which would employ smart monitoring, control and self-healing technologies [29].

Regardless of a mass roll-out of microgrids, it is apparent that smarter control techniques could be used to prevent catastrophic events, such as the 2003 U.S. blackout [17], from ever occurring. Even with phasing in microgrids, new control techniques are necessary to aid in coordination of islanding and reconnecting these networks with the main grid [57]. At the moment, techniques for microgrid reconnection are limited to manual synchronization of voltage, angle, and frequency between a Point of Common Coupling (PCC) [64, 4, 58]. These methods assume a microgrid is connected to the main grid at one location and fail to address the potential inability to directly synchronize said measurements. Due to necessary synchronization, generation/load may be cut or energy storage may aid in achieving satisfactory conditions/synchronization [5]. New techniques may need to address a wider range of measurements from potentially multiple PCCs as well as sub-optimal reconnection times due to the potential inability to synchronize. Said measurements may come from different sources, potentially direct device measurement or high accuracy estimates from state estimation.

State estimators require large amounts of redundant and accurate data [6], fortunately with the availability of faster and larger measurement sets, network controls may be achievable in real time and help create a more robust network in the future. Methods of monitoring the electrical grid have sprung up in forms of Supervisory Control and Data Acquisition (SCADA) and most recently Phasor Measurement Units (PMUs). PMUs allow time synchronized data at high sample rates aiding in many control methods and state estimations [15, 16]. The ability to estimate the state of an electrical network is a powerful tool, allowing correcting actions to take place in response of contingencies and sub-optimal network operation. With the introduction of near real time monitoring, automatic control techniques become feasible for networks in a plethora of situations. These control techniques may prove invaluable in preventing situations ranging from poor network operation to full scale blackouts. As measurement explosion occurs, it becomes difficult to develop well understood rules when building intelligent controls. Artificial

Intelligence (AI) and Machine Learning (ML) may aid in creating these network controls when faced with non-trivial actions in certain situations. As of now, it is important to address the fact that full coverage of PMUs may not be feasible in the foreseeable future, thus placements of said measurements must be carefully considered. Much research exists in optimizing PMU locations [7, 28, 27, 41, 69], however said optimal locations may depend on the application stemming from the usage of said measurements.

When leveraging these measurements, the possibility of adversarial manipulation of said data must be addressed. With PMUs making use of GPS synchronization, an adversary may steer the original signal away causing inaccurate measurements [32]. The electrical grid may be improved with faster and more intelligent control, however the operation of said controls with manipulated measurements may do more harm than good to the network in certain situations. As a result, pure automated control algorithms may be limited to mainly guide network operators in their actions. Other techniques may make use of pre-processing measurement data to decrease the potential of malicious data entries.

As the electrical power system moves towards a Smarter Grid, it is important to prepare for the potential changes of grid architecture and operation. With that being said, utilities are beginning to deploy their own microgrids [48]. With the operation of both microgrids and subnetworks, it is imperative that adequate control schemes are developed to aid in a Smarter Grid. The avoidance of both sub-optimal network operation and failure is imperative. With the availability of smarter monitoring, the ability to better develop smart controls becomes feasible. The improvements are not without concerns; it is important that the community addresses the increased attack platforms available to adversaries on the cyber side as future automated monitoring/controls will rely heavily on network communications [8, 30, 25].

1.1 Thesis Focus

This research focuses on introducing a new control technique for reconnecting microgrids and avoiding situations in which a reconnection may create network instability. As microgrids are phased into the electrical grid, it is important that the ability to stably reconnect said microgrids exists with high confidence. An automated approach will aid operators in real time when tasked with deciding when to reconnect a subnetwork.

Furthermore, security concerns are raised in conjunction with PMUs and smart meters. Impacts of malicious PMU data on the previously proposed control technique for microgrid reconnection are raised. The consequence of smart meters directly manipulating loads throughout a network is also demonstrated to show how a well designed attack may cripple a network. Solutions to identifying malicious PMU data entries are discussed and adopted to further increase the robustness of future control techniques using said measurements.

A policy based solution for network control is also explored. When attempting to mitigate system damage in face of failures, discrete protective elements may operate to save specific components or alleviate network loading. We explore the addition of network operator actions to aid against a failing network in the form of policy rollout. With this, a policy (set of actions) can be determined to attempt to save the network in an online setting.

1.2 Thesis Layout

The research is organized by first discussing the paper: Load Oscillating Smart Meter Attack in Chapter 2. This paper focuses on the ability to hack into a large number of smart meters and consequently turn loads on and off creating an oscillatory based attack. We follow up with Chapter 3 which is composed of the paper: A Learning Scheme for Microgrid Reconnection. This demonstrates a potential control scheme that allows the reconnection of a microgrid in the face of sub-optimal conditions and potentially adversarial PMU data. Chapter 4 further explores impacts of adversarial PMU measurements and methods of identifying malicious data. A policy based approach to preventing blackouts is discussed in Chapter 5. We conclude the thesis with the implications of our work and potential future work.

Chapter 2: Load Oscillating Smart Meter Attack

The following chapter is composed of the published paper [35]. This focuses on the impacts of cyber based attacks on smart meters within a network.

2.1 Abstract

This paper investigates the potential impacts of load oscillating attacks in a microgrid to the stability of the main power grid. The adversary is assumed to be able to control switches within compromised smart meters and thus is able to dynamically connect or disconnect the corresponding loads within the microgrid. Using the commercial PSS/e time-domain simulator with the IEEE Reliability Test System (RTS-96), we demonstrate the impacts of attacks cycling the total load of the microgrid. Cycling attacks with different load oscillation frequencies and magnitudes are considered. We found that for certain oscillation frequencies, oscillating 30 percent of the total microgrid load can significantly harm the main grid stability.

2.2 Introduction

The power network is trending towards a smarter and more intelligent entity due to developments in *smart grid* over the past several years. These advancements occur at the transmission, distribution, and consumer levels. Distribution networks have seen a multitude of developments including communication system upgrades, automation of distribution elements, load control, and Advanced Metering Infrastructure (AMI) [1]. With such improvements, the potential to dynamically control and protect distributed networks becomes more feasible. Unfortunately with the broadening of said improvements, new attack surfaces are introduced to the power grid [47]. For instance, attackers may intrude into AMI and manipulate the data or inject false control data in order to remotely control switches. This paper focuses on studying whether such attacks launched at a distribution network can affect the main grid stability.

Monitoring of the distribution level has been difficult in the past, but with the introduction of smart meters in AMI, the ability to dynamically track load details becomes possible. Smart meters not only provide power system operators with real-time information of individual customer load (*e.g.*, single house load) but also allow operators to remotely control switches in order to connect or disconnect individual loads [56]. Such two-way communication creates a new security concern because compromised smart meters (or compromised channels between smart meters and the operators) may not only cause the leak of measurements, but also allow the adversary to connect or disconnect the corresponding customer loads [2].

There have been reported successful hacking of smart meters allowing one to sniff data or even inject commands into the device. The ability to control the devices and shut down power is a real possibility and may have harmful outcomes pertaining to wide grid stability [30]. The diverse ways to hack smart meters could be as simple as reverse engineering one or using software radio programmed to mimic communication devices to learn how to communicate with the meter. Compromised meters could be used to spread *malware* to other smart meters allowing easier accessibility for smart meter based attacks for adversaries [25]. The spreading of software with malicious intent has already been tested and successfully carried out by researchers in which a worm was created and traveled through other meters [42]. As a result, it is possible that a few compromised smart meters could lead to a large network of compromised meters.

Such security risks of meters are becoming more concerning due to the deployment of such systems outpacing security efforts [43]. Such deployments stem from sources such as the Smart Grid Recovery Act in which $4.5 billion dollars were directed toward modernizing the power grid. These changes are occurring very quickly. In 2014, the U.S. had 58,545,938 AMI installations with 88% being residential customer installations [3]. It is expected that the number of smart meters installed worldwide will grow from 313 million, in 2013, to nearly 1.1 billion in 2022 [53]. With the rapid growth of such network based systems and lack of research on such security risks, major consequences may occur from compromised systems.

The immediate thought of smart meter attacks would seem to be price fraud in which meters are tampered with allowing setting changes. In 2009 many reports of such fraud were reported in Puerto Rico where utility employees changed meter settings such that customers were charged less [25]. Main security research in regards to smart meters have

focused on privacy or fraud, however more intelligent based attacks could create more serious consequences, *e.g.,* disrupting control of power grid [8]. Other attacks may rely on oscillating portions of the grid to disrupt power delivery. Such attacks have been developed and tested on small test cases as seen in [39], [40]. These coordinated attacks may have significant impact on grid stability depending on the source of said switching.

In this paper, we focus on attacks that exploit compromised smart meters in order to introduce load oscillation *within* a single microgrid. In general, perturbation at a distribution level is considered to have a negligible impact and is ignored in the main grid control. The potential impacts of *elaborately designed* perturbation at a distribution level on the main grid stability have not been well understood. This paper aims to fill this gap by providing case studies with load oscillating attacks.

2.3 Attack Model

In this paper, we consider an adversary who compromised a subset of smart meters within a microgrid and is capable of controlling switches associated with them. A straightforward attack could involve a one time dropping of the entire adversarial load, however this may not capture the worst case result as the network may recover from the single instance as opposed to a more intelligent attack.

With a more intricate attack, an adversary may choose to cycle loads to confuse the system and create potential problems with protective device operation. In this case, the cycling would pertain to actual load manipulation, not just the meter readings. With a cycling attack, the adversary may control loads and switch them on and off at a frequency potentially harmful to the network. With such an attack, network convergence issues may result from component stress or protective schemes occurring to aid in the current cycled state whilst being detrimental to the next cycled state. Consistent cycling may result in compounded consequences leading to instability. As a result, this attack would change the actual operating point of the network.

We modeled such an attack in a microgrid setting. With the RTS-96 case [67], we assumed the third zone to be our microgrid. We set up an attack model by choosing all loads in the microgrid to be susceptible to attacks. All initial load models on buses are split into 10 individual feeder representations with identical values. We choose the amount of load the microgrid may cycle and pick adversarial feeders for each bus based

on this information. For example, if we allow 50 % of the total microgrid load to be cycled, we represent this by cycling 5 adversarial feeders at each bus in the microgrid. The adversarial feeders for each bus are chosen at random and independently to ensure diversity. We perform attacks that vary in the set of adversarial feeders, the cycling frequency, and the attack duration. This attack is performed when the microgrid is interconnected to the main grid to determine the impact of load oscillation within the microgrid on the main grid.

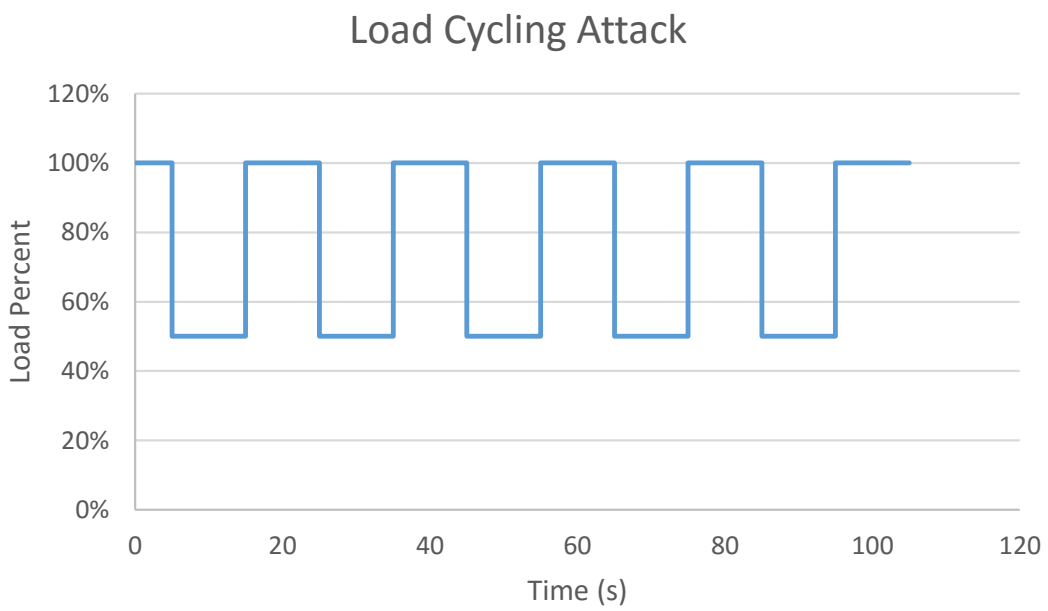

Figure 2.1: Example microgrid attack cycling half of the load

An example attack on the microgrid is shown in Figure 2.1. This attack controls 50 % of the microgrid load and cycles them at a frequency of 0.05 Hz. The attack lasts for 100 seconds with loads oscillating from 50 % to 100 % of the microgrid load. It is important to note that the absolute total load of the grid may decrease as the attack goes on because some uncompromised feeders can be disconnected due the protective scheme triggered by load oscillation.

For clarity, an example is described and the expected outcomes are discussed. We use the attack seen in Figure 2.1. The network begins with the main grid and microgrid

interconnected and operating at the precomputed steady state. At 5 seconds the adversarial loads are switched off in the microgrid leading to only 50 % of the microgrid load remaining. The network will then be exposed to transients and attempt to converge to a new stable point of operation. Relays will operate if their thresholds are exceeded for a pre-set time. These relays are discussed in more detail in Section 3. Relay operation may result in load shedding, line tripping, or machine tripping. The attack will then restore the adversarial loads at 15 seconds. It is important to note that the load shedding from the protection scheme does not discern between adversarial and non-adversarial feeders. The load shedding may shed adversarial load even when it has been switched off. As a result, when an adversarial load is restored, it may no longer be served. The attack may then decrease in magnitude as adversarial load is shed throughout the attack period.

2.4 Methodology

A commercial dynamics simulator, PSS/e, was used to conduct experiments on the RTS-96 case. Due to the RTS-96 case being a set of three identical networks with two extra buses for interconnection of said zones, we selected the third zone to represent our makeshift microgrid. In order to adequately represent the the individual loads and the associated feeders at a distribution level, we broke each load model presented in the test case into 10 identical individual loads, each of which is connected to the substation by a different feeder. An example would be a 100 MW, 10 Mvar load on a certain bus. The load is broken into ten loads each with values of 10 MW and 1 Mvar.

In order to represent adequate cycling behavior, we allowed feeders to be either adversarial or non-adversarial. We first select the amount of load we wish to cycle in the microgrid, we then use this to determine how many adversarial feeders exist per bus. The representation was as follows: If we were to cycle load from between 40% and 100% of the total load, each bus in the microgrid would have 4 non-adversarial feeders and 6 adversarial feeders. It was important to create diversity throughout the case, thus we chose the adversarial feeders at each bus uniformly at random and independently. For each amount of load we cycle, we choose the adversaries at each bus as we stated before. For simulations using the same amount of load cycling, we use the same distribution of adversaries throughout the microgrid; this is to ensure that the same amount of load being cycled at different frequencies will result in different behavior due to this change

in frequency, not due to the change of adversarial locations.

Dynamics were implemented in the case by using salient generator models along with the IEEE type 1 exciter and IEEE type 2 governor. Protective relays were also built for the case which included overcurrent line relays, undervoltage + underfrequency bus relays, and underfrequency machine relays. The initial pickup points for overcurrent line relays were synthesized by running the steady state solution and using the line currents at hand. The relay pickup time was chosen to be 140 % of the operating current in the steady state with a zero reset time of 5 seconds. Line relays would trip the associated branches if they timed out during operation. Table 2.1 shows the other setpoints for the line relays.

Table 2.1: Different operating points for overcurrent relays

	Percent of Pickup	Trip Time (s)
Point 1	100 %	5
Point 2	120 %	0.2
Point 3	140 %	0.15
Point 4	160 %	0.1
Point 5	180 %	0.05
Point 6	200 %	0

Undervoltage and underfrequency load shedding protection was produced by placing relays at each feeder previously created. For the ten feeders per bus, five different setpoints were created to represent 20 % load shedding at a bus per setpoint; these are shown below in Table 2.2. In order to create variability in load sheds pertaining to frequency, we introduced four different types of setpoints that represent time until load shed. The time until operation for frequency points is a set value divided by a random variable, x, that can take on a value of 1, 2, 3, or 4. The relays for the ten feeders in the same bus shared the same value of x. This allows more diversity across bus relay configuration ensuring not all loads are shed at once due to common frequencies in smaller islands. Voltage points do not need such variability as their voltages differ enough throughout the network.

A similar technique for machine relaying was used to ensure diverse frequency trips. We attach three underfrequency trip points for each machine in the case and introduce

Table 2.2: Undervoltage/Underfrequency load shedding relays operating points

	Volt Pickup	Trip (s)	Freq Pickup (Hz)	Trip (s)
Pt 1	0.88 P.U.	3	59	4/x
Pt 2	0.85 P.U.	1	58.5	2/x
Pt 3	0.80 P.U.	0.5	58	1/x
Pt 4	0.75 P.U.	0.25	57.5	0.5/x
Pt 5	0.70 P.U.	0.1	57	0.25/x

random time trips as shown in Table 2.3. We use a random variable, y, that can take on values of 1, 2, or 3.

Table 2.3: Points of operation for generator protection relays

	Frequency Pickup (Hz)	Time Until Trip (s)
Point 1	58.5	y
Point 2	57.5	y/2
Point 3	56	y/4

As stated earlier, a feeder can be adversarial or not. The protective scheme is setup such that adversarial loads may be shed even when cycled off. This adequately models an operator shedding a feeder during protective actions even if the feeder has been completely shut off by an adversary. We assume that the operator does not know the exact distribution of load on the bus, thus feeders are shed according the the predesigned protection schemes.

2.5 Results

We performed tests on the RTS-96 case by allowing all buses in the microgrid to have a number of adversarial feeders. We tested on cases that cycled 30 %, 50 %, and 80 % of the microgrid load. We also cycled each attack at frequencies of 0.05 Hz, 0.1 Hz, 0.5 Hz, and 1 Hz. Attacks lasted for 100 seconds with the simulation terminating 75 seconds after the attacks end.

Interestingly, we observed that the oscillation of more load in the microgrid did not

Table 2.4: Remaining load, machines, branches after an attack, and amount of buses lost

Attack	Active Load (MW)	Reactive Load (MVAR)	Machines	Branches	Lost buses
Normal Operation	9037	1737	99	120	0
80 % at 0.05 Hz	4201	845	57	49	38
80 % at 0.1 Hz	4745	901	66	62	26
80 % at 0.5 Hz	6220	1121	83	74	14
80 % at 1 Hz	6884	1259	90	77	13
50 % at 0.05 Hz	6454	1174	90	83	12
50 % at 0.1 Hz	6381	1157	89	75	13
50 % at 0.5 Hz	7097	1302	90	79	13
50 % at 1 Hz	7052	1299	90	76	13
30 % at 0.05 Hz	5685	1016	82	59	27
30 % at 0.1 Hz	3953	793	55	45	41
30 % at 0.5 Hz	6509	1260	86	86	5
30 % at 1 Hz	6741	1294	91	80	4

always correspond to the worst outcome; in fact we found that oscillating from 70 % to 100 % of the loads (*i.e.*, cycling 30% of the microgrid load) at 0.1 Hz in the microgrid caused a major loss in load, machines, buses and branches. Table 2.4 shows the remaining load after attacks lasted for 100 seconds.

The loads remaining are those that exist at stable islands after an attack occurs. Unstable islands are either directly disconnected due to protection or not counted as served if the island has not converged upon termination of the case. Similar behavior can be seen with the remaining machines after attacks shown by Table 2.4.

Normally high frequency-oscillation of feeders did not adversely effect machine tripping too much. We saw that low amplitude oscillation did not always result in less machine tripping. The cycling of half the loads seems to result in fewer machines tripping than cycling either 80 % or 30 %. We also observe a similar story for the remaining branches/transformers after each attack.

In Table 2.4, we track remaining load after an attack along with machines (generation), and branches (in service two winding transformers and bus tie lines). The remaining branches after attacks seem highly correlated with the remaining machines in the case. We see that the cycling of 50 % of microgrid loads results in less machine trips on average through differing frequency attacks. The remaining branches and

transformers only represent ones that exist in stable islands upon completion of a test case. Normally high-frequency load oscillation caused only a small fraction of machines and branches to trip, however lower frequency coupled with high or low load oscillation removed a large portion of branches and transformers in the working case.

We also observe how many buses were lost after an attack ended in Table 2.4. The number of lost buses came from two different scenarios. The first cause was due to protective line tripping which isolated buses into separate islands. If an island were to lose all machines, the buses are set out of service due to the inability to serve as an active portion of the case. The other cause is due to islanding, however the island becomes unstable before losing all generation. If an island becomes unstable and reaches a point in which it cannot converge, the entire island is set out of service. The least number of buses that are disconnected come from a high frequency-oscillation, low amplitude-oscillation attack. We observe cycling half of the load in the microgrid seems to have less variability in terms of all parameters shown in Table 2.4 with respect to frequency, whilst frequency has a big impact on cycling 30% and 80% of the load. The worst case scenario again results from oscillating 30 % of the load at a frequency of 0.1 Hz.

One major result that was not immediately expected was the large impact small oscillations of load could cause as opposed to medium oscillations. The worst performance outcome was found when cycling only 30 % of the load at 0.1 Hz. We found that low amplitude oscillation allowed the perturbation to be felt by a large portion of the grid before protective islanding isolated the attack. As a result, the low oscillating attack at 0.1 Hz was able to cause enough distortion to cause a large island to become unstable before it broke into protected regions. In case of high-amplitude oscillation attack, the protective load shedding scheme was able to isolate the attacked region in the microgrid, however the microgrid and connected buses normally did not survive due to such an aggressive attack. The moderate-amplitude cycled load attack normally caused protection to isolate the fault, but some portions of the attacked microgrid still survived due to less drastic cycling. It is important to note that different protection schemes may result in differing behavior among the explored attack scenarios. The assumed testing did not account for an adversary knowing the protective layout of the system; as a result, more sophisticated attacks may cause further damage to the grid (in particular bypassing known protective operations isolating the attack).

2.6 Conclusion

This paper investigated potential impacts of cyber attacks that exploit compromised smart meters to oscillate the total load of a microgrid. We found that an intelligent adversary could produce a small-amplitude load oscillation at a problematic frequency that can distort the grid and cause protective measures to take actions resulting in many losses and islanding. With presented material with respect to smart meter security exploitation, possible attacks on such systems could create harmful consequences that need to be addressed. No countermeasures to such attacks were explored in this paper, but remain a focal point for future research.

Acknowledgment

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780.

Disclaimer

Chapter 3: A Learning Scheme for Microgrid Reconnection

The following content in this chapter is pending publication in IEEE Transactions on Power Systems. It describes a method for finding stable points in which a sub-networks may reconnect to a main grid. The authors consist of: Carter Lassetter, Eduardo Cotilla-Sanchez, and Jinsub Kim.

3.1 Abstract

This paper introduces a potential learning scheme that can dynamically predict the stability of the reconnection of sub-networks to a main grid. As the future electrical power systems tend towards smarter and greener technology, the deployment of self sufficient networks, or microgrids, becomes more likely. Microgrids may operate on their own or synchronized with the main grid, thus control methods need to take into account islanding and reconnecting of said networks. The ability to optimally and safely reconnect a portion of the grid is not well understood and, as of now, limited to raw synchronization between interconnection points. A support vector machine (SVM) leveraging real-time data from phasor measurement units (PMUs) is proposed to predict in real time whether the reconnection of a sub-network to the main grid would lead to stability or instability. A dynamics simulator fed with pre-acquired system parameters is used to create training data for the SVM in various operating states. The classifier was tested on a variety of cases and operating points to ensure diversity. Accuracies of approximately 85% were observed throughout most conditions when making dynamic predictions of a given network.

3.2 Introduction

As we make strides towards a smarter power system, it is important to explore new techniques and innovations to fully capture the potential of such a dynamic entity. Many large blackout events, such as the blackout of 2003, could have been prevented with smarter

controls and better monitoring [18]. Phasor measurement units, or PMUs, are one such breakthrough that will allow progress to be made in both monitoring and implementing control to the system [51]. PMUs allow for direct measurement of bus voltages and angles at high sample rates which makes dynamic state estimation more feasible [15, 16]. With the use of PMUs, it is possible to improve upon current state estimation [52] and potentially open up new ways to control the grid. The addition of control techniques and dynamic monitoring will be important as we begin to integrate newer solutions, such as microgrids, into the power network. With these advanced monitoring devices, microgrids become more feasible due to the potential for real-time monitoring schemes. The integration of microgrids bring many benefits such as the ability to operate while islanded as well as interconnected with the main grid; they provide a smooth integration for renewable energy sources that match local demand. Unfortunately the implementation of microgrids is still challenging due to lacking experience with the behavior of control schemes during off-nominal operation.

Currently, microgrids are being phased in slowly due in part to the difficulty of operating subnetworks independently as well as determining when they can be reconnected to the main grid. Upon reconnection of an islanded sub-network to the main grid, instability can cause damage on both ends. It is important to track instabilities on both the microgrid and main grid upon reconnection to accurately depict the outcome of reconnection. Works in the literature have focused on the potential of reconnecting microgrids to the main grid, in particular aiming at synchronizing the buses at points of interconnect with respects to their voltages, frequencies, and angles [64, 4, 58]. Effort has been directed at creating control schemes to minimize power flow at the point of common coupling (PCC) using direct machine control, load shedding, as well as energy storage, to aid in smooth reconnection [34, 10].

In some cases we may need to look at larger microgrids or subnetworks in which multiple PCCs exist. In such scenarios, it becomes much more difficult to implement a control scheme that satisfies good reconnection tolerances in regards to minimizing bus frequency, angle, and voltage differences at each PCC. In addition to the possibility of multiple PCCs, it is possible that direct manipulation of the system becomes limited, compromised, or unsupported with respect to synchronization. In order to address these shortcomings, we implement an algorithm that dynamically tracks and makes predictions based on the system states, providing real-time stability information of potential

reconnections.

Due to the complexity of the power grid, it is difficult to come up with a verbatim standard depicting the potential stability after reconnection of a subnetwork. With advances in the artificial intelligence community, we can make use of machine learning algorithms in order to explore vast combinations of sensor inputs, states, and control actions. This can be done in a similar fashion to successful techniques applied to other power system problems as seen in the research literature [19, 33, 54, 66]. In this paper we propose to use a machine learning algorithm, specifically a Support Vector Machine, to predict safe times to reconnect a portion of a grid. The Support Vector Machines allow one to build a classifier predicated upon training data by determining a linear separator in a specific feature dimension [20]. As seen in [70] we can create a knowledge base consisting of training and testing data using an appropriate power system model and simulator. Diversity of data points in the knowledge base can be achieved by incorporating load changes allowing multiple operating points [70, 54]. Simulators have been used prevalently to create data and work has been performed to show the agreement between different simulators [61]. As a result, we will assume the creation of data for our technique is adequate upon diligent modeling.

In the proposed machine learning approach, PMU measurements are used as input features that will be used by a learning algorithm to predict which class the features belong to, either stable or unstable reconnection. As of now, PMUs are not as prevalent in the system to assume full state observability in real time, thus it is important to take into consideration limited PMUs when implementing techniques [71]. This paper borrows the concept of electrical distance which suggests voltage changes propagate adhering to closeness of buses [22, 23]. As a result, without getting into the PMU placement optimization problem, this paper assumes that PMUs were located nearby the PCCs.

The proposed method leverages real-time PMU data to predict system stability upon reconnection. PMUs make use of GPS synchronization [32] which can create an attack platform for adversaries by changing or shifting the time synchronization. Use of erroneous or compromised PMU data could lead to incorrect predictions that would degrade system stability due to hidden failures that remain dormant until triggered by contingencies [63]. We demonstrate a potential framework that can make accurate predictions in face of partially compromised PMU data.

It is important to highlight the reasoning behind introducing a learning based approach to the problem as previous methods dealing with synchronization exist. By leveraging techniques similar to a synchro-check relay [19, 33, 54, 66], it is possible to become confident of a stable reconnect for systems even in the dynamic domain. Said technique focuses on limiting key measurement differences in voltages, angles, and frequencies between a select PCC in a connecting network. However, it is difficult to set proper thresholds on the voltage, angle, and frequency differences, below which we allow reconnection. Too low thresholds may cause many opportunities for stable reconnection missed, while too high thresholds may lead to unstable reconnection. Further, thresholds for different relays may have to be set differently as sensitivity changes for different locations. In addition, such a reconnection strategy limits the reconnection decision for certain tie line to depend only on the tie line measurements thereby rendering the decision possibly suboptimal. The proposed learning scheme provides an integrated framework that takes into account all the aforementioned challenges including the following:

- The challenge of setting up proper decision regions is naturally handled in the training phase of the learning scheme.
- Our prediction of stable reconnection timing for certain tie line relies on datastream from diverse PMUs, not limited to those associated with a single tie line.
- The learning scheme improves upon the synchro-check relay scheme in a sense that the possible decision rules of synchro-check relays are included in the collection of decision rules to be considered by our learning scheme, for most choices of learning methods.

The proposed technique is not without its flaws. While the learner does a good job improving on being less restrictive on PMU locations and can provide a better confidence interval for stability, it is associated with required computation time. The learner needs to be fed unique data based on the network at hand in order to see improvements on the previous methods. Using the learner in a real-time environment is trivial, however the actual training of said learner needs careful consideration along with a unique skill-set. If the learner is correctly set up it could become a potentially powerful tool for determining real-time stability of network reconnection.

The contributions of this paper are as follows. We propose a machine learning framework to learn a classifier that can predict the stability of potential reconnections of a sub-network regardless of the number of PCCs. The proposed scheme is evaluated using the RTS-96 case and the Poland case and demonstrates high classification accuracy, around 90%. We demonstrate the scalability [36] of the proposed scheme using the Poland case; the amount of required computation scales reasonably as the network size grows. Lastly, we present that the proposed scheme can succeed for a large-scale grid even when only a few PMUs are available for use. This implies that the proposed scheme is feasible even when the number of trustworthy PMUs is quite limited.

The remainder of this paper is organized as follows. Section II gives a brief background of Support Vector Machines (SVM). Section III covers the methodology to create a power system classifier. Section IV discusses results from experiments with the proposed algorithm. Section V provides the conclusions.

3.3 Problem Formulation and Preliminaries

In this section, we formulate stability prediction of microgrid reconnection as a machine learning problem and provide the preliminaries describing the SVM. While we describe the problem formulation in the context of SVM, the proposed framework is applicable to generic machine learning approaches.

We propose to leverage SVM to predict stable reconnection timings of a microgrid based on real-time PMU measurements. Conceptually, the SVMs transform an input feature vector into a higher-dimensional space and applies a linear classification rule to predict its class label [12].

In our context, real-time measurements collected from PMUs at certain time point form an input feature vector. The input vector is associated with a binary class label, either 1 or -1, depending on whether reconnection of the microgrid at the current time would lead to a stable operating point or an unstable point, respectively. We assume that there exists an unknown conditional probability distribution that characterizes the conditional distribution of the true class label given an input vector. Under this assumption, we will use the SVM framework to learn a classifier that maps input vectors to true class labels with high probability. The learned classifier can be used in practice to predict the consequence of a reconnection when certain PMU measurements are observed.

In order to learn a classifier, we need training data consisting of a number of input vectors $x_1, ..., x_n$, and their associated class labels $y_1, ..., y_n \in \{-1, 1\}$. The methodology to obtain the training data will be explained in Section 3.4. Given a set of training data, the SVM uses a basis function, denoted by $\phi(\cdot)$, to map input vectors into a higher-dimensional space in order to enhance linear separability. The SVM takes these feature vectors as inputs with their corresponding labels and is trained with the information. Specifically, a separating affine hyperplane is obtained by solving the following primal problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \tag{3.1}$$

subject to

$$y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \ i = 1, 2, ..., n \tag{3.2}$$

where the regularization term with the parameter C penalizes the training data points that are on the wrong side of the margin. The solutions w^* and b^* to the above optimization define the SVM classifier as follows:

$$f(x) = \text{sign}[(w^{*T} \phi(x) + b^*)] \tag{3.3}$$

where the offset b^* is derived from the dual solutions[12].

The example in Fig. 3.1 depicts a classifier built for prediction of two classes. In this example, the squares represent one class and the circles the other. The separating hyperplane is found by solving the optimization problem (3.1), with margins existing for

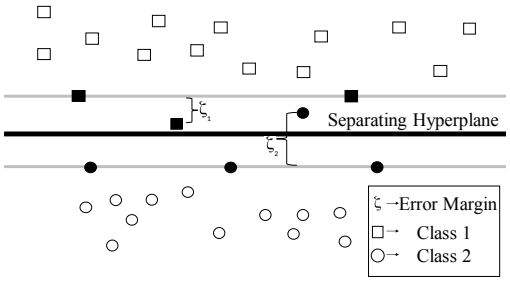

Figure 3.1: Example representation of decision and error boundaries for a Support Vector Machine

each class. The support vectors, seen in bold, are examples closest to the margins. A solution may not always have classes completely separated; the penalty will be associated to the distance past the margin, ζ_i, and the weight, C.

In the case that the dimension of $\phi(x_i)$ is significantly higher than that of x_i, solving the dual of (3.1) can lead to an alternative expression of the classifier that is substantially easier to compute. The dual of (3.1) is:

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{3.4}$$

subject to

$$y^T\alpha = 0, \quad 0 \leq \alpha_i \leq C, i = 1, 2, ..., n \tag{3.5}$$

where α_i denotes the Lagrangian multiplier for the ith constraint of (3.5), and e denotes a vector of all ones. In the dual formulation, the basis function $\phi(\cdot)$ is integrated into the matrix Q by the use of a kernel function $K(x_i, x_j) \triangleq \phi(x_i)^T\phi(x_j)$. Specifically, the (i, j) entry of Q is equivalent to $y_i y_j K(x_i, x_j)$. Many kernels exist, but the most relevant one used in this paper is the Radial Basis Function (RBF), or Gaussian, kernel shown below:

$$K(x_i, x_j) = e^{(-\gamma|x_i - x_j|^2)} \tag{3.6}$$

where γ is the hyperparameter to be optimized via cross-validation. The solutions of the dual problem provide an alternative expression of the classifier (3.3):

$$f(x) = \text{sign}\{\sum_{i=1}^{n}(\alpha_i y_i K(x, x_i) + b)\} \tag{3.7}$$

Using the above expression has computational advantages over the use of (3.3), because $K(x, x_i)$ is in general easier to compute than $w^{*T}\phi(x_i)$. This is true for kernels with the dimension of $\phi(x)$ being significantly larger than x such as the RBF kernel. Further, the majority of weights, α_i, will be zero; only the support vectors will have nonzero weights.

3.4 Training the SVM using a Dynamic Simulator

In this section, we present the machine learning framework for predicting stable reconnection timings of a microgrid as well as the detailed procedure to train the classifier with a power system dynamic simulator. As suggested earlier, we train the SVM to predict the stability of reconnection for a microgrid when certain PMU measurements are observed. In order to train the SVM, we need to first acquire a set of training examples, each of which is a pair of an input vector (i.e., a vector of PMU measurements) and the true class label (i.e., stability of reconnection when the input vector is observed as PMU measurements). Unfortunately, it is difficult in practice to obtain sufficient training data from realistically sized power systems as obtaining a pair requires disconnecting and reconnecting the microgrid. Thus, we resort to leveraging a power system dynamic simulator to create training data by running a variety of scenarios for the target system.

3.4.1 Overview

Fig. 3.2 illustrates the procedure that we follow for the experiments in this paper. We begin this procedure by breaking up a test case into different operating points. Each of these operating points are used to create different initial conditions unique to their operating point. These new initial conditions are built by randomly scaling the load throughout the network. We perform dynamic simulations consisting of islanding and reconnecting the microgrid to create our synthesized PMU measurements and determine the stabiliy of said reconnection. After gathering the data, we break our data into training and testing sets which are used to train the classifier. We then use the classifier to monitor PMU streams and predict the potential stability of a microgrid reconnection.

We chose the 73-bus version of the IEEE Reliability Test System (RTS-96) [68] and the 2383-bus version of the Poland Test Case as test cases for evaluation of our approach as they are well tested in the community [72]. The RTS-96 provides a convenient topology to implement and test islanding, whereas Poland serves as a larger network to more closely model a practical system. For the RTS-96 and Poland case we used the procedure described below to create several operating points. The Poland test case used a modified winter peak snapshot to ensure diverse data could be gathered during the creation of different initial conditions.

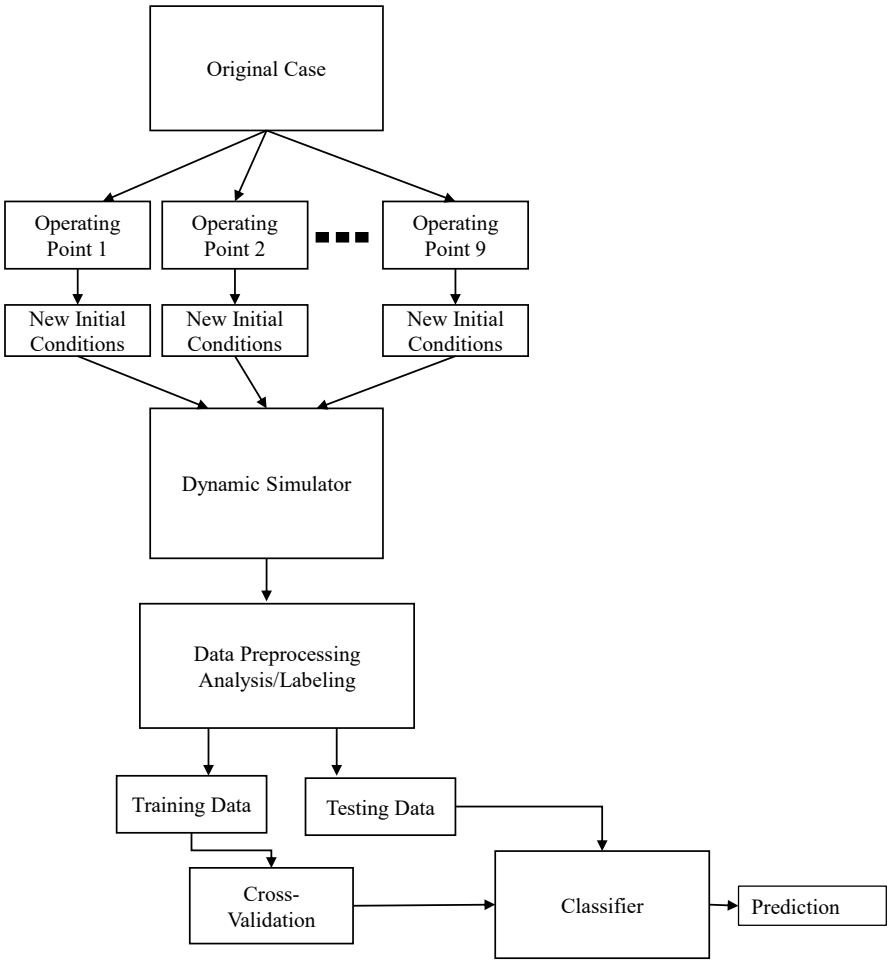

Figure 3.2: High level overview of the process to create a classifier

We began with a specific network and created different operating points by uniformly changing load locations throughout the network. Loads were also uniformly scaled at random when building these new operating points. We then simulate the dynamics of the system with Siemens PTI PSS/e and perform the islanding and reconnection scenarios. Upon completion of simulations, bus voltages and angles before the reconnection of islands are used as features and the outcome of the case (stable or unstable) are used to label the set. The raw data produced are separated into training and testing sets in which cross-validation is performed exclusively on the training set to build an adequate

classifier.

3.4.2 Diversifying Operating Points

It is important to take into account test cases that can reproduce various operating points depending upon, for example, time of the day, day of the week, or season [55]. In this way, the classifier will be useful for a diverse set of network states. We created different operating points by shuffling and scaling loads at random throughout the system. Upon obtaining the new demand distribution, we ran a steady state solution of the case and considered it stable and usable if the voltage magnitude set was between 0.9 p.u. and 1.1 p.u. for the RTS-96 case or 0.8 p.u. and 1.1 p.u. for the Poland case. For each operating point we created different initial conditions by changing active and reactive loading on each bus, according to Eqs. (3.8) and (3.9):

$$P_{\text{new}} = P_{\text{old}} + \theta P_{\text{old}}, \quad \theta \sim U(-a, b) \tag{3.8}$$

$$Q_{\text{new}} = Q_{\text{old}} + \gamma Q_{\text{old}}, \quad \gamma \sim U(-a, b) \tag{3.9}$$

where P_{new} and P_{old} denote the new system active power and original system active power, respectively; Q_{new} and Q_{old} denote the new system reactive power and original system reactive power respectively. For scaling, θ and γ are independent and identically distributed random variables that are uniformly distributed in $[-a, b]$.

3.4.3 Dynamic Simulation

We are interested in the interaction between the sub-network and main grid upon reconnection. In order to observe the main reconnection mechanisms, we simulate the power system dynamics with a time-domain simulator software (Siemens PTI PSS/e) along with a custom built command line interface (Python API[1]). We first used a research-grade dynamic simulator alongside PSS/e to cross-validate and tune the dynamic machine models [49, 62]. The dynamic models selected consist of salient machines for the generators, IEEE Type 1 exciters, and IEEE Type 2 governors. We initiate each simulation run in PSS/e with a flat start check in order to ensure the dynamic models do not alter

[1]Application Program Interface

the steady state solution and also that no protective elements are operating during the steady state. We added relay models and protection schemes to our test cases, including overcurrent, undervoltage, and underfrequency relays. The overcurrent relays are set up using the line limit standard data that come with the selected test cases. We configured load-shedding, line-tripping, and generator disconnection actions for undervoltage and underfrequency situations. During the dynamic simulation we monitor bus voltages, angles, and frequencies.

For each initial condition obtained for a given operating point of the original test case we run a dynamic simulation. After the initialization period we proceed to island a pre-defined portion of the test case in which the two isolated systems run independently for a certain amount of time. The sub-network is then reconnected with the main grid and continues to run until the end of our simulation time.

3.4.4 Data Generation and Labeling

The proposed learning scheme necessitates the collection of training examples which will be used for training an SVM classifier. We exploit the aforementioned dynamics simulation module and various initial conditions to create diverse training examples. As stated earlier we create different operating points for our test cases, and we then create new initial conditions for each operating point for diversity. Each initial condition case will give us a single feature vector along with a single label. The feature vector for each case consists of the bus voltages and angles measured by PMUs at the time point before reconnection. Angles were unwrapped to the first turn, between -180 and 180 degrees. We assume that the PMU set is fixed for clarity.

The label for each initial condition case represents whether the case became stable or unstable upon reconnection of the sub-network to the main grid. Labeling was done based on the PSS/e convergence monitor which would alert the Python interface if the network did not converge at any point in time. If the API observes the 'network not converged' message, we assume immediately that the PSS/e was unable to solve the differential-algebraic system of equations and label the case unstable. We added additional convergence rules during labeling which allowed more cases to be labeled unstable if voltage collapses, there are very large oscillations, divergence or intolerable frequency spikes occurred. If the case satisfied the rules of stability we provided, it

was labeled as stable. We store all case data in the form of their feature vectors and associated class labels.

3.4.5 Test Scenarios

We split the full data set into two subsets; one representing the training set to create the classifier, and the other one to test the accuracy of the classifier. Three main methods of creating the training and testing set were used and described below.

3.4.5.1 Single Operating Point Case

To assess the baseline capability of the classifier we start with the simplest case by assuming our classifier is trained and tested on examples originating from a single operating point. We create the training and testing sets with a single operating point. The different initial conditions from said operating point will be the only examples populating the training and testing sets. The created sets will be used independently from other operating points. This test proves the ability for the classifier to make predictions with PMU data streams coming from a well known network operating point.

3.4.5.2 Multiple Operating Points Case

To build on the previous test, we use multiple operating point to form training/testing sets for our classifier. We previously demonstrate a method to test individual operating points, however a more universal classifier would leverage all available data from different operating points. This test allows a more generalized baseline accuracy to be derived. This can be achieved by mixing the initial conditions from all available operating points from the full data set. We create the training set by randomly selecting a subset of the mixed data. The remaining unselected data is placed into the testing set. This suggests that the classifier may be trained on a set consisting of examples from different operating points and make predictions on different examples from the same operating points.

3.4.5.3 Unseen Operating Points Case

It is important to assess the ability of the classifier in the face of unknown operating points. The baseline accuracies to be produced from the previous test scenarios implies predictions would rely on the network having a finite set of most common operating points. It can be assumed that larger networks would create an exponential number of potential stable operating points. It is necessary to show that large networks could adopt the proposed technique by populating the testing set with examples from unknown operating points unseen in the training set. Unlike the last test scenario, we keep the different initial condition data from each operating point separate. We create a random subset of operating points that will be used to populate the training set with their different initial conditions. The initial conditions created from the remaining operating points are then put into the testing set. The exclusion of certain operating points from the training set ensures that the classifier must make predictions on a testing set that contains only examples from unobserved operating points. The unknown operating points represent potential distributions of load in the network that are unaccounted for in training, but may still exist at any given time.

3.4.6 Classifier

Given the prepared training and testing sets, the next step is to define and build the classifier. As stated earlier, it may be necessary to remap the features to another dimension in which classification is easier, this leads us to choose from different kernels and hyperparameters. SVM is very sensitive to the kernel and hyperparameters chosen, thus it is important to setup the classifier in a way that maximizes our prediction accuracy. In order to find optimal kernel and hyperparameters, we use k-fold cross validation on the training set [45]. Random oversampling is employed to balance the training set such that the classifier will not be over-fitted to the majority class[65].

The next step is to train the classifier with the entire set of training data available. Upon completion of training, the classifier is able to make predictions of classes for unseen input feature vectors. Specifically, the classifier predicts whether the system it has been trained on will be stable or unstable if it were to reconnect at the given time. We made use of the Python library *scikit-learn* [50], which includes implementations of

machine learning algorithms such as SVM.

3.5 Results

In this section, we present the performance of the proposed method for predicting stability of microgrid reconnection. For evaluation, we used first the RTS-96 test case to demonstrate the approach and the Poland case to benchmark the methodology against a real sized power system [14]. As stated previously, the proposed classifier can account for multiple PCCs in a network. For example, due to the choice of islanding Zone 3 in the RTS-96, we consider the two PCCs shown in Fig. 3.3.

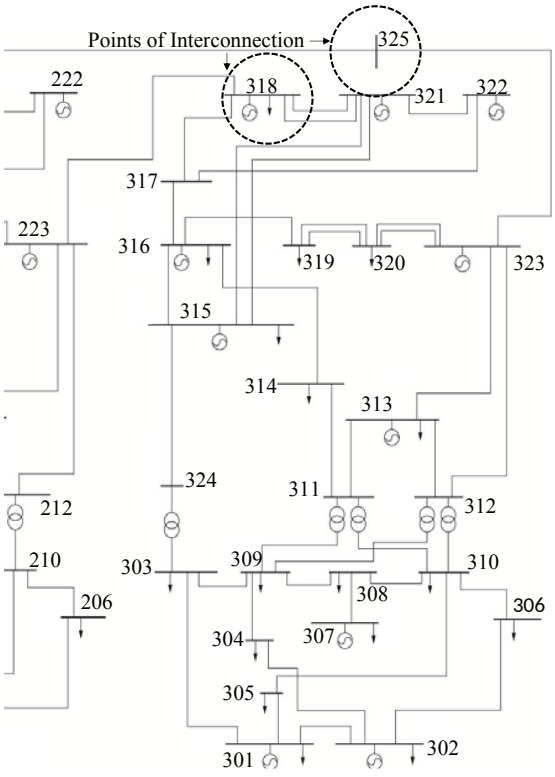

Figure 3.3: Points of interconnection in the RTS-96 case.

3.5.1 RTS-96

For the RTS-96 case we created nine different operating points and gathered 400 different initial conditions for each. The RTS-96 case is made up of three sub-networks that are mostly identical to one another, and we chose to island Zone 3 which contained bus numbers in the 300s. The intentional islanding occurred five seconds into the simulation, the reconnection event occurred at 45 seconds, and we terminated the simulation at 120 seconds. We did not implement protection schemes for this baseline scenario. We leveraged data from all buses in the RTS-96 case to test the classifier to begin with. These buses were chosen due to their proximinity to the PCCs.

3.5.1.1 Single Operating Point Case

We began by creating a classifier for each operating point and observed the accuracy attained on each class. For each operating point we chose 100 cases of class stable and 100 cases of class unstable to train the classifier. We applied 10-fold cross validation to the training data to find optimal kernel and hyperparameter values. From these we observed the best performance was achieved with the RBF kernel along with a specific set of hyperparameters. Some operating points had differing hyperparameters when their classifiers were built. As such, Table 3.1 shows the selected hyperparameters for each operating point.

We observed that training and testing on individual operating points yields results that suggest some are easier to predict than others. The worst case operating point can predict unstable cases with an accuracy of 80%, as seen in Table 3.2, however most other operating points can make predictions at a much higher accuracy. In Table 3.2, Class 1 accuracy and Class 0 accuracy represent the probabilities of detecting stable reconnections and unstable reconnections correctly, respectively. It isn't feasible to assume a system will be operating with one specific load distribution which is why multiple operating points were introduced. At the same time, the operating point's load distributions were created semi-stochasticaly in the sense that loads were introduced to value changes consistent with equations (3.8) and (3.9) and randomized, but still had to satisfy the voltage p.u. stability requirements. These distributions ensured operating points were different enough that it would cover a a case in which the system operates with high

Table 3.1: Classifier setup after cross-validation

Operating point	Kernel	γ	C
1	RBF	0.000001	100
2	RBF	0.0001	10
3	RBF	0.000001	10
4	RBF	0.000001	10
5	RBF	0.00001	1
6	RBF	0.0001	10
7	RBF	0.00001	1
8	RBF	0.000001	10
9	RBF	0.00001	0.1

randomness, which is harder to make predictions for than most systems.

3.5.1.2 Multiple Operating Point Case

We also investigated a universal classifier that assumes an operator would not have immediate access to detailed knowledge of the current operating point of the system. With this assumption, we create a universal classifier training it with the training set consisting of cases from all nine operating points, 100 stable and 100 unstable cases from each operating point. The reason for training with the same number of stable and unstable cases is to prevent a classifier from being potentially being skewed based on the priori of the class distributions in the training set. Similarly we use 200 cases from each operating point to ensure no operating point dominates the classifier during training.

We performed the aforementioned cross-validation technique and obtained the best classifier, which is an RBF kernel with a γ value of 0.00001 and a C value of 1. We tested it on the test set, and the results are shown in Table 3.3. The accuracies when jointly trained perform relatively well as a whole, however some operating points can result in difficult to classify examples. We kept the operating points separate to observe how well the universal classifier does on each particular case and then obtained the average accuracy over the whole test set to demonstrate overall performance.

Table 3.2: Accuracies for RTS-96 operating points independently trained

Operating point	Class 1 accuracy [%]	Class 0 accuracy [%]
1	97.8	100
2	80	99.2
3	90.7	97.1
4	97	80
5	84.7	89.6
6	91.3	85.9
7	89.5	86.0
8	96.7	90.6
9	90.6	81.3
Average	90.9	90

3.5.1.3 Inference with trustworthy PMUs

We investigated the performance of the proposed method when only a small subset of PMUs are used for classification. We created a small subset of PMUs to choose from located at buses: 118, 121, 218, 221, 223, 318, 321, 323, 325. It turned out that using a smaller subset of PMUs does not substantially degrade performance if the subset is properly chosen. Among the assumed PMU locations, we selected a PMU to be allowed in the trusted subset only if they were immediately adjacent to a PCC in the network. As a result we can choose a handful of desired PMUs to be used. Out of these PMUs, for this experiment we only selected either two or three to be secure, then we trained and tested on the smaller subset. Table 3.4 illustrates the results of this experiment, whereby Class 1 represents a stable reconnection and Class 0 represents an unstable reconnection.

The main reason for obtaining better results with limited PMUs in some test cases is due to the exclusion of PMUs that are either adding noise to the classifier or not providing relevant information. A higher number of features leads to the need for more training data to create an adequate classifier. If we use PMUs that do not provide useful information, building the classifier becomes difficult with limited training data. We observe that it may not be feasible to produce large quantities of training data which can lead to better results from subsets of PMUs rather than the entire set. This is shown

Table 3.3: Accuracies for RTS-96 operating points jointly trained

Operating point	Class 1 accuracy [%]	Class 0 accuracy [%]
1	86.8	100
2	97	72
3	79.2	99.4
4	90.1	82.9
5	89.4	88.7
6	100	78
7	91.2	88.8
8	95.6	92.5
9	93	74
Average	91.4	86.3

in randomly chosen subsets in Table 3.4 for RTS-96 as well as in the following section for the Poland case.

The above results suggest that the proposed method can be adjusted to be resilient to potential cyber attacks that may manipulate part of PMU data. In the event that the integrity of PMU measurement data is not fully guaranteed due to cyber threats[46], we cannot rely on the classifier processing the full set of PMU measurements. To effectively handle such a case, we can prioritize protection of a certain small subset of PMUs such that the integrity of their measurements can be strongly guaranteed even in the presence of cyber adversaries. Our results imply that if the trusted subset is properly chosen, the classifier can perform with high accuracy based on the trusted PMU measurements.

3.5.2 Poland Network

For the Poland case we created twenty-four different operating points and generated roughly 240 different initial conditions total. On top of the steady state diversity implemented, we obtained data from 50 reconnection points spanning randomly between 40-55 seconds from each initial condition to implement more temporal diversity. We incorporated a protective scheme by adding overcurrent relays on each transmission line, as well as undervoltage and underfrequency relays on each bus. We allowed relay oper-

Table 3.4: Accuracies for RTS-96 operating points with subsets of trusted PMUs

PMU location [bus number]	Class 1 accuracy [%]	Class 0 accuracy [%]
118, 318	92.7	87.6
118, 321	92.8	87.6
121, 318	92.4	87.5
118, 121	90.1	85.6
323, 325	92.2	86.8
218, 321, 325	93.4	87.2
221, 223, 323	94.1	86.6
121, 218, 318	93.2	87.3
118, 121, 218	90.2	86.0
318, 323, 325	94.3	86.4

ation to trip lines, shed load, or disconnect generators. The overcurrent relays were set based upon the transmission line limits from the original test case. Table 3.5 provides an overview on the relay configuration.

Table 3.5: Overcurrent relay configuration

Point	Pickup [%]	Trip time [sec.]
1	100	5
2	125	0.2
3	137.5	0.15
4	150	0.1

Underfrequency and undervoltage relays were used for bus and generator monitoring and protection. Setting the voltage thresholds is straightforward given the baseline variability of voltages for each bus. Frequency variability is more challenging to set up without obtaining more information from the operation of a large network. Thus, we grouped buses with similar frequency response and introduced different frequency threshold points throughout the system. As a result, load shedding and generator tripping due to underfrequency events allowed for heterogeneous disconnection, generally a

more accurate depiction of system survival in a real case. Synthesized time-dial points for underfrequency bus relays were setup as shown in Table 3.6, depicted by rows (LS). For generator relays, a random value ($y = \{1, 2, 3, 4\}$) was chosen and scaled for the time-dial points shown in Table 3.6, depicted by rows (GR).

Table 3.6: Undervoltage/underfrequency load shedding (LS) and under/over frequency generator (GR) relay configurations

Point	Pickup volt. [p.u.]	Trip [sec.]	Pickup freq. [Hz.]	Trip [sec.]
LS 1	0.92	5	49.5	5, 4, 3, 2
LS 2	0.88	0.5	49	2, 1.5, 1, 0.5
LS 3	0.75	0.2	48.5	1, 0.75, 0.5, 0.25
GR 1	-	-	48.5, 51.5	y
GR 2	-	-	47.5, 52.5	$y/2$
GR 3	-	-	46, 54	$y/4$

Table 3.7: Baseline Poland network accuracies

PMU location [bus number]	Class 1 accuracy [%]	Class 0 accuracy [%]
Unseen Operating Point Case	94.4%	96.0%

Since the Poland test case is divided by default into five zones, we solved the steady state of the case when islanding certain zones. Zone 5 was a good candidate for intentional islanding due to a low mismatch for generation and demand, as well a voltages within acceptable operating limits, thus it was selected to be the sub-network of interest in this experiment. During the dynamic simulations we islanded the sub-network at 2 seconds. We implemented a more temporal approach with respect to reconnection to capture real-time changes in the network. As a result, reconnection times ranged from 40-55 seconds for each dynamic simulation. Unlike the RTS-96 experiment, we did not assume full PMU coverage of a large scale network to begin with. We only allowed a PMU on a bus if it is immediately attached to the interconnection between the sub-network and the main grid. We were left with 30 available PMUs in the Poland network to build a feature vector. Since each PMU contains a voltage and angle measurement the dimension of the feature vector is 60 (if using the entire set of PMUs). As we stated

Table 3.8: Unseen operating point case accuracies for Poland network with subsets of trusted PMUs

PMU location [bus number]	Class 1 accuracy	Class 0 accuracy
2218, 171, 118, 335, 2249, 214, 126, 139, 125, 303, 174, 2226, 186, 1607, 165, 1761	94.3%	95.6%
186, 2331, 315, 139, 167, 10, 2234, 2124, 225, 2218, 2226, 178, 125, 2249, 126, 1607	95.8%	95.3%
303, 2234, 2124, 315, 225, 335, 10, 118, 140, 2226, 2218, 214	88.4%	96.3%
2218, 140, 174, 126, 125, 118, 2234, 171, 2124, 15, 167, 139	94.6%	95.6%
167, 139, 214, 335, 178, 2226, 315, 118	89.8%	96.1%
174, 2249, 2218, 118, 2331, 1607, 141, 166	95.6%	95.5%
139, 165, 2218, 2226	96.0%	95.1%
127, 2249, 118, 166	96.3%	94.3%

earlier in the procedure description, the next step was to create labels based on the convergence of the case. Figures 3.4 and 3.5 illustrate labeling examples for stable and unstable cases, respectively.

Figs. 3.4-3.5 depict frequencies of two buses on either side of an interconnection point. One can observe that case labeled as stable case exhibits a reconnect where the frequency signals converge to a common operating state. The unstable case shows the frequency of Bus 126 spike and immediately flat-line representing a bus trip. As described in the methodology section, if the network did not converge, it would have immediately been labeled unstable. The rules of stability in the Poland case additionally enforced that at least 2370 of the 2383 buses in the case were in service after reconnection of the island.

We partitioned the 722 different initial conditions in accordance to the two test cases described in Section III-E: multiple operating point case and unseen operating point case. For each test case, we used 10-fold cross validation together with random oversampling to learn optimal hyperparmaters and train the classifier (see[45, 65] for details of these methods). The set aside test set was then used to determine the classifier's accuracy.

The baseline accuracies of the Poland network are seen in Table 3.7. The unseen operating point case represents the case in which the testing set contains data from

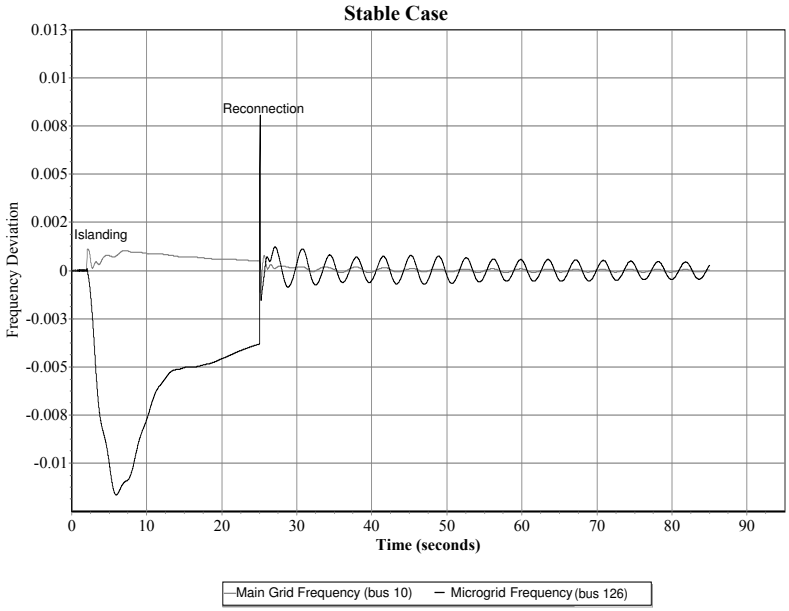

Figure 3.4: Stable reconnection of Poland microgrid and main grid.

operating points that do not exist in the training set. With the unseen operating point test, the proposed algorithm demonstrated over 90% accuracy. In particular, the results from the unseen operating point case suggest that our classifier can demonstrate this accuracy even when the classifier is trained based on a few operating points and tested for an *unseen* operating point case. This implies that the proposed method is scalable and suitable for use in a large-scale grid; the classifier does not have to be trained for all possible operating points, and training with a few suffices. As stated earlier for the smaller test case experiments, we also investigate the accuracy of the classifier for a scenario when the system is compromised. As a response, our classifier makes use of a trusted set of PMUs and makes predictions based on their measurements. A variety of subsets from the available PMU full set make up our possible trusted scenarios, as shown in Table 3.8. The results indicate that some subsets still perform well even in the face of unknown operating points.

In the larger Poland case it seems more prevalent that decreasing the amount of features can lead to similar performance to the full set. The adoption of this control technique would bring into question whether a utility could provide enough training

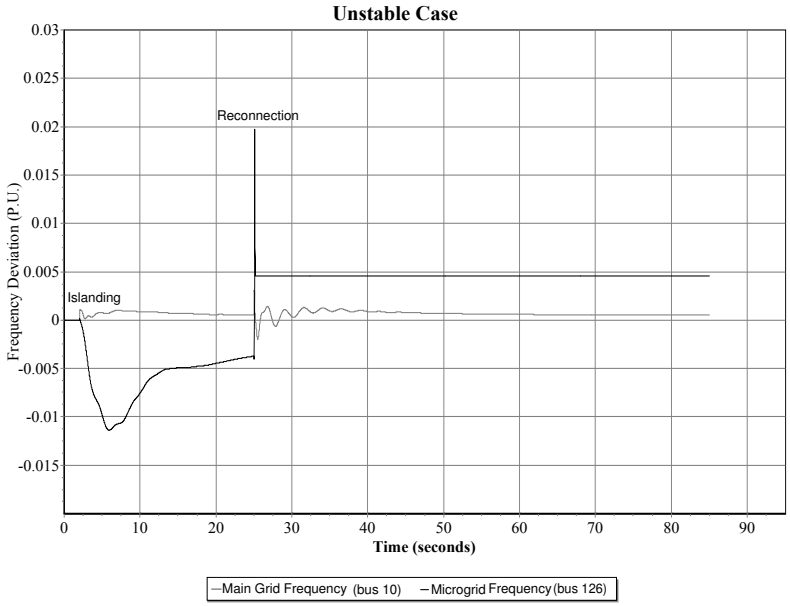

Figure 3.5: Unstable reconnection of Poland microgrid and main grid.

data, specifically the number of training examples, for the classifier. If limited training data is provided, the usage of an optimal subset of PMUs instead of the entire available set could yield adequate accuracies. Indeed it is always interesting to observe that less amount of information give similar results. It is explained in this case by considering a high dimension of the feature space. For high dimensional feature vectors, it is difficult to learn an accurate classifier with limited amount of training data. Utilities with the ability to archive and make available relatively large amounts of training data could still make use of a a large set of PMUs, if available, and potentially observe higher accuracies with respect to the quantity of training examples provided in these experiments.

3.6 Conclusion

This paper presents a machine learning approach for the prediction of stable reconnections of a power system sub-network. The proposed approach leverages a power system dynamics simulator to generate synthetic, yet realistic in terms of size, training examples that are subsequently employed to train a classifier. The interactions between power sys-

tem dynamics and protection mechanisms are complex, and the exact derivation of an optimal control strategy is not always feasible. However, as demonstrated in this paper, a machine learning approach can be useful to capture many unintuitive behaviors and make predictions in real-time based on PMU measurements. Future improvements on the training aspect may be necessary as the procedure to build said classifier is relatively sophisticated and requires in depth knowledge. The method may not be directly usable by operators as a result of the necessary understanding to build a well trained classifier. The classifier was tested on a variety of cases and operating points to ensure diversity. Accuracies of approximately 90% were observed throughout most conditions when making dynamic predictions of a given network. Existing work in literature is limited to the dynamic realization of reconnection stability, however future work may leverage said technique in a more time sensitive way. In addition, cyber attacks on PMUs in a subset may distort the classifier thus creating the need to implement techniques on verifying the authenticity of the data streams.

Acknowledgment

This material is based upon work supported by the Department of Energy under Award Number DE-OE0000780.

Chapter 4: Security Concerns for PMU Based Controls

4.1 Introduction

With the introduction of smarter monitoring and controls, the assumption of reliable data may not always hold. This chapter focuses on the pre-processing of PMU data to minimize potentially corrupt data from impeding correct operation of developed control schemes. We make use of said tools with conjunction of the developed Learning Scheme proposed in Chapter 3, however the pre-processing could be extended to other control techniques making use of PMU measurements. Figure 4.1 shows an overview of the steps used to pre-process PMU data.

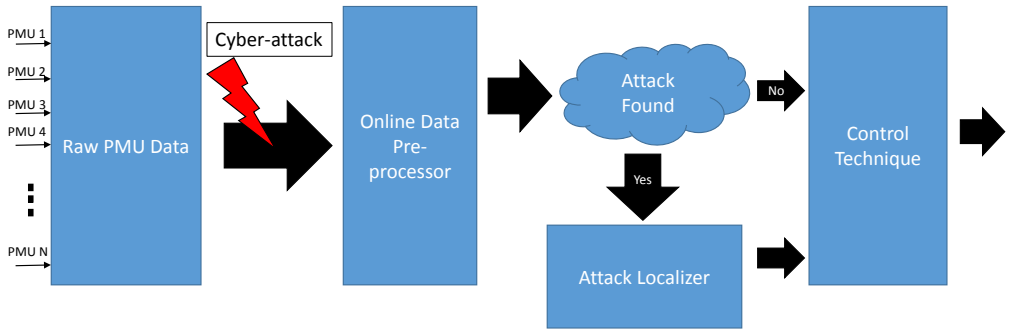

Figure 4.1: High level abstraction of detecting and localizing malicious PMU measurements

4.2 Raw PMU Data

We start with discussing the measurements used and practical assumptions to be made. If we assume the deployment of PMUs to be confined to a local area, we may make use of the fact that the power system is slowly varying. The measurements from buses within close proximity will swing in a similar manner allowing the argument that the time measurements from multiple PMUs will be low rank [24]. A low rank matrix can be seen as having high linear dependence between rows and columns. When making this assumption, one must ensure that the PMUs are grouped close to one another and in the same network. If PMUs between two isolated systems are merged together, the rank may dramatically change and impact future analysis of the data. With regards to the Learning Scheme proposed in Chapter 3, two main sets of PMUs exist where 15 would reside on the microgrid side and 15 on the main grid.

PMU measurements are synthesized in the same way as Section 3.4.4 from the Poland test case. We make use of measurements from the 30 key PMU locations before reconnection for a given window of time. As stated before, the microgrid and main grid are operating independently from one another when islanded, thus two sets of measurements are created to allow separate analysis.

4.3 Cyber Attack

With the PMU measurements at hand, we introduce the potential of an adversary corrupting certain PMU devices. We make a key assumption that an adversary will have limited resources and access to a small portion of the available PMUs in the set. The limitation must be made to ensure the PMU measurement matrix is not corrupted to the point of destroying much of the linearity seen between PMUs. The attack can be simplified to three main matrices, X, or the ground truth PMU measurements, A, the attack matrix, and $\tilde{X}$, the observed matrix after attack. This combination can be seen as a basic addition in equation (4.1).

$$X + A = \tilde{X} \tag{4.1}$$

A simplified example of this is shown in equation (4.2) which represents an attack on row 2. In this case, the data is formated with rows representing each PMU and columns

being each time step. The first goal would be to localize the attack in an effort to eliminate any corrupt PMUs. Building off localization, the ability to recreate the attack matrix and similarly the true measurement matrix may be useful to increase the usable data if time allows. With time sensitive control schemes it is desirable to quickly localize corrupt PMU devices to throw away said data and leverage only trustworthy PMUs.

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ x_{31} & x_{32} & \dots & x_{3n} \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \dots & 0 \\ A_{21} & A_{22} & \dots & A_{2n} \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} \tilde{x_{11}} & \tilde{x_{12}} & \dots & \tilde{x_{1n}} \\ \tilde{x_{21}} & \tilde{x_{22}} & \dots & \tilde{x_{2n}} \\ \tilde{x_{31}} & \tilde{x_{32}} & \dots & \tilde{x_{3n}} \\ \tilde{x_{m1}} & \tilde{x_{m2}} & \dots & \tilde{x_{mn}} \end{bmatrix} \tag{4.2}$$

4.4 Attack Localizer

The ability to ensure only trustworthy PMUs are used in control schemes is of paramount concern to avoid any malicious impacts to a network. Previous literature makes use of PMU measurement low rank by implementing the popular nuclear norm minimization [9]. Said technique in its most basic form is seen in equation (4.3).

$$\min ||X||_* + \lambda||A|| \qquad s.t. \qquad \tilde{X} = X + A \tag{4.3}$$

This method is used to optimize the solution finding the sparse attack matrix (A) and the low rank of 'true' PMU measurements (X). The nuclear norm $||X||_*$ operator finds the sum of the singular values of matrix X which impacts the rank.

With relatively small windows of data, the nuclear norm minimization approach is appealing when tasked with finding adversarial PMUs due to the ability of being relatively quick. Upon finding a PMU that seems to have been targeted, the user may disregard the data when making network decisions and avoid potentially misguided recommendations from PMU based control schemes. Another benefit of the nuclear norm minimization is the innate ability to reconstruct both the attack matrix and ground truth measurements. The whole reason localization is possible is due to the method directly finding the values of the attack matrix.

The main concern of the nuclear norm minimization based approach is with respect to how well the data at hand adheres to the assumptions. If an adversary has the ability to

control the entire set of PMUs, this approach becomes easily exploitable. Furthermore, if PMUs within an analyzed set do not swing well together, there exists the potential of poor localization. It should be stressed that this pre-processing should be limited to that of which PMUs are grouped well with one another and only a limited set are adversely impacted in a given time frame.

4.5 Online Data Pre-processor

In some control schemes, it may be important to determine if an attack has occurred as fast as possible. The nuclear norm method may be usable, however it does have a period of delay due to the direct estimation of the attack and ground truth measurement matrices. An online data pre-processor would be leveraged to predict whether an attack on the measurements has occurred. This in turn would eliminate the necessity of continually using the nuclear norm method to find attacks.

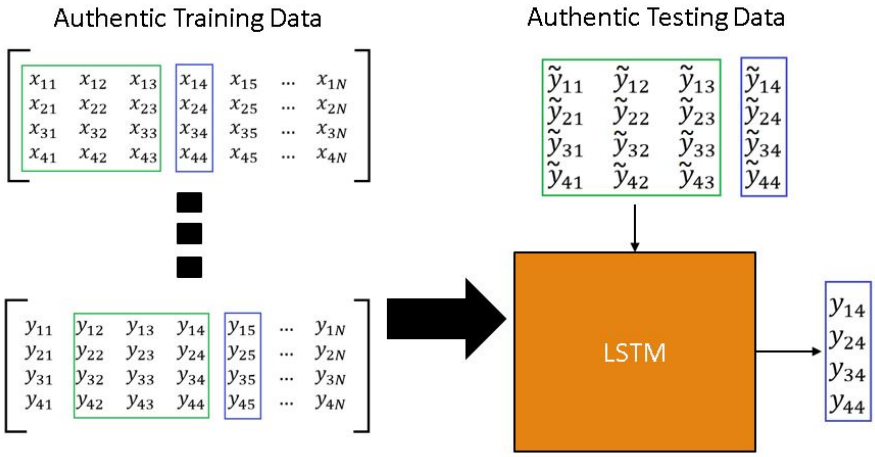

Figure 4.2: Training LSTM network.

The most basic approach would be to implement a screening algorithm that would solely detect if an attack is occurring. A more complex solution would be to detect an attack and localize it without necessarily directly reconstructing the attack and ground truth measurement matrices. The next solution is seen to be the most time consuming where the attack and ground truth measurement matrices are directly solved to localize

the attack, such as the nuclear norm approach.

Different solutions for this step were attempted, however were unsuccessful as general solutions. The most interesting involves the use of Deep Learning (DL) where a Long Short Term Memory (LSTM) network was used temporally to predict the next time step's measurement set from a previous set of measurement time steps. The basic idea is shown in Figure 4.2 in which the measurements highlighted in green represent the features and the time step measurements in blue are the targets. The LSTM is trained to make predictions of the measurements for the next time step. With a real time stream of data, one can compare the LSTM estimate to the data coming in. Since attacks on measurements may come in many different forms, the LSTM must be trained with attack free measurements. The testing data will also be attack free and used to build a rule that discerns measurements as potentially corrupt or not.

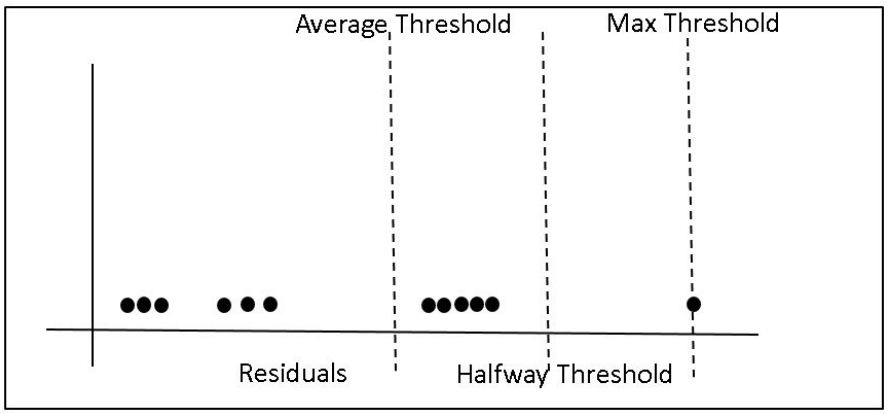

Figure 4.3: Testing set residuals and potential thresholds.

The residual between the real time incoming data stream and predicted measurements from the LSTM may help build the rule. We make use of the testing data to create a distribution of residuals. An example of these residuals can be seen in Figure 4.3. The true target $\tilde{T}$ and estimated target T from Figure 4.2 are used to create a residual depicting how well the LSTM can predict the next time step measurements. This residual contains the difference between all features estimated at the next time shown in equation (4.4)

$$T = \begin{bmatrix} y_{11} \\ y_{21} \\ y_{31} \\ y_{41} \end{bmatrix}, \ \tilde{T} = \begin{bmatrix} \tilde{y_{11}} \\ \tilde{y_{21}} \\ \tilde{y_{31}} \\ \tilde{y_{41}} \end{bmatrix}, \quad R = \sum_i |T_i - \tilde{T}_i| \tag{4.4}$$

With a distribution of residuals, one may choose a threshold to act as a 'rule' when predicting if a measurement matrix contains malicious data. For example, if we choose the threshold to occur at the maximum observed residual seen in our test set, Figure 4.3, then any predicted target from the LSTM that exceeds that residual will be flagged as malicious. It can be seen that the higher the threshold, the higher the probability of false negatives occurring. If our threshold is chosen to be a smaller residual, then we run the risk of having more false positives.

This approach yielded positives results when leveraging measurements from limited operating points. Unfortunately when increasing the amount of operating points, the LSTM's ability to provide accurate predictions decreases. As a result, the residuals of predictions increases which degrades performance when predicting if a measurement matrix is malicious or not. Due to this, the approach would be severely limited to predictions on a well known operating point of a given network.

4.6 Security Risks with No Preprocessing

We show a quick example of an angular based attack on PMU measurements used in our previously created Learning Scheme from Chapter 3. This serves to demonstrate the potential degradation of control schemes when faced with malicious data. The attack is limited to a particular PMU subset comprised of buses: 127, 2249, 118, 166. We observe the impacts of shifting the measured angle of a single PMU bus within the subset at a given time. Table 4.1 shows the results of introducing said angle shifts to certain PMU measurements and feeding them into the classifier.

It can be seen that the attack significantly degrades the classifier's performance for the given subset of PMUs. The accuracy of class 1 and class 0 are shown for each attack magnitude and location pair. As a result, the addition of pre-processing is necessary to ensure targeted PMUs are thrown out and a secure set of PMUs may be located and used when performing the classification. Further implications show that any other

Table 4.1: Angular shift attack on Subnetwork Reconnection Learning Scheme

PMU Targeted/ Angular Attack	PMU: 127 (Class 1, Class 0)	PMU: 2249 (Class 1, Class 0)	PMU: 118 (Class 1, Class 0)	PMU: 166 (Class 1, Class 0)
5	(94.8%, 93.9%)	(95.8%, 95.1%)	(95.9%, 94.5%)	(95.7%, 94.3%)
15	(83.3%, 92.4%)	(55.1%, 97.0%)	(57.2%, 96.6%)	(56.8%, 96.8%)
30	(51.9%, 89.3%)	(40.3%, 97.9%)	(48.4%, 97%)	(47.7%, 97.2%)
-5	(94.1%, 94.6%)	(96.0%, 94.6%)	(95.4%, 94.7%)	(95.6%, 94.8%)
-15	(75.0%, 93.6%)	(56.8%, 96.5%)	(55.5%, 97.1%)	(56.0%, 96.9%)
-30	(43.3%, 90.7%)	(49.2%, 96.7%)	(42.2%, 97.8%)	(45.0%, 97.7%)

control method leveraging secure PMU subsets can make use of this technique to ensure their security.

4.7 Results with Preprocessing

After showing the implications of an attack on our previous control technique, we make use of the pre-processor to limit any malicious data from having any impact on our classifier. We show results in terms of false positives where the pre-processor determines a PMU device to be incorrectly adversarial and false negatives where the pre-processor determines a PMU device to be incorrectly trustworthy. The measurements are synthesized similar to that described in Section 3.4.4. We break the analysis into two windows containing measurements from 15 PMUs on the microgrid and 15 PMUs on the main grid. Tables 4.2 and 4.3 show the results of implementing the nuclear norm minimization technique on a set of randomly attacked PMUs in a given group of PMUs. We look at 198 different sets of PMU measurement windows to test this approach.

Table 4.2: Analysis on 15 PMUs located near interconnection on main grid

Attack Type	False Negatives	False Positives
1 PMU, U(-0.01,0.01)	0/198	2/198
2 PMU, U(-0.01,0.01)	0/198	2/198
3 PMU, U(-0.01,0.01)	0/198	2/198

The attacks are a trivial uniform scaling of each time point measurement for the given attacked PMUs. We limit ourselves to observing the attack of up to 3 targeted PMUs and scaled by up to 1% of the original measurement values. For the most part,

Table 4.3: Analysis on 15 PMUs located near interconnection on microgrid

Attack Type	False Negatives	False Positives
1 PMU, U(-0.01,0.01)	0/198	119/198
2 PMU, U(-0.01,0.01)	0/198	117/198
3 PMU, U(-0.01,0.01)	0/198	107/198

this particular attack is well identified on the main grid side with limited false positives. The microgrid side seems to have a difficult time with false positives, mainly due to bus 2218 which is consistently classified as a targeted PMU. This PMU seems not to swing well with the other PMUs within the group making it difficult to analyze, as a result it is consistently labeled as malicious. It is important to note that out of the 198 data windows, there are 15 PMUs in each. The loss of one PMU in each analysis may not be such a big deal and may be worth the elimination to ensure the analysis catches the actual targeted PMUs.

Chapter 5: Policy Based Network Control

5.1 Introduction

We now focus on the topic of developing smarter control schemes to aid the electric power network when faced with contingencies. Contingencies normally relate to failures in the network such as faults, line trips, generator disconnection, and many others. These contingencies may in turn impact other portions of the network creating a cascading failure. To demonstrate this, the RTS-96 case is included in Figure 5.1 along with two arbitrary contingencies.

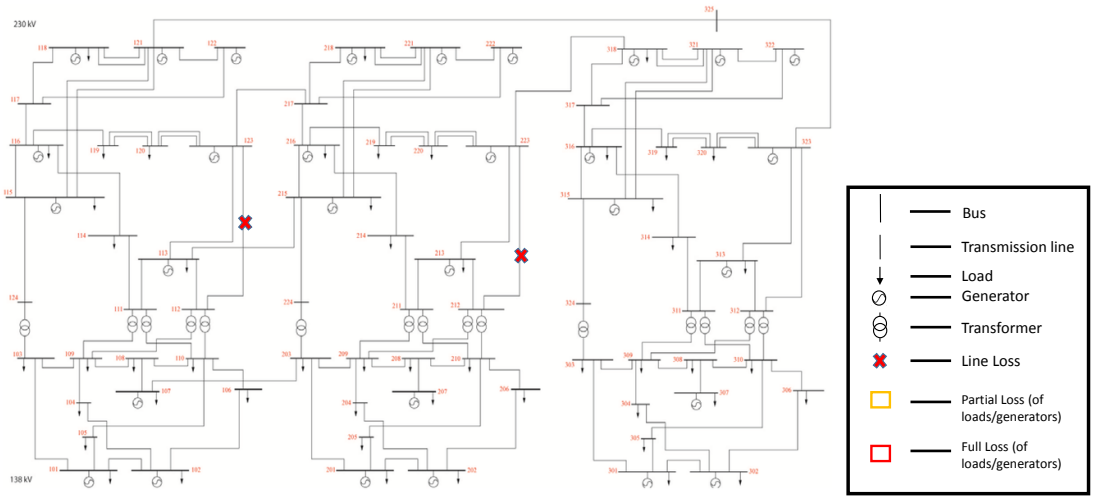

Figure 5.1: Two line contingencies on the RTS-96 case.

With the introduction of two contingencies, in this case two line trips, other components may become stressed to the point that other protective elements operate. As a result, the line trips between buses 112-123 and buses 212-223 will cause other lines to

pick up the slack and transfer the electricity that the two previous lines were previously handling. It is apparent that other lines may become overloaded and trip which could potentially lead to cascading failures propagating throughout the network. We can see in Figure 5.2 an example of the potential of the start of cascading failures. Depending on the protection in the system, cascading failures may occur in a different manner. For this example, we assume that all lines, transformers, buses and machines are protected.

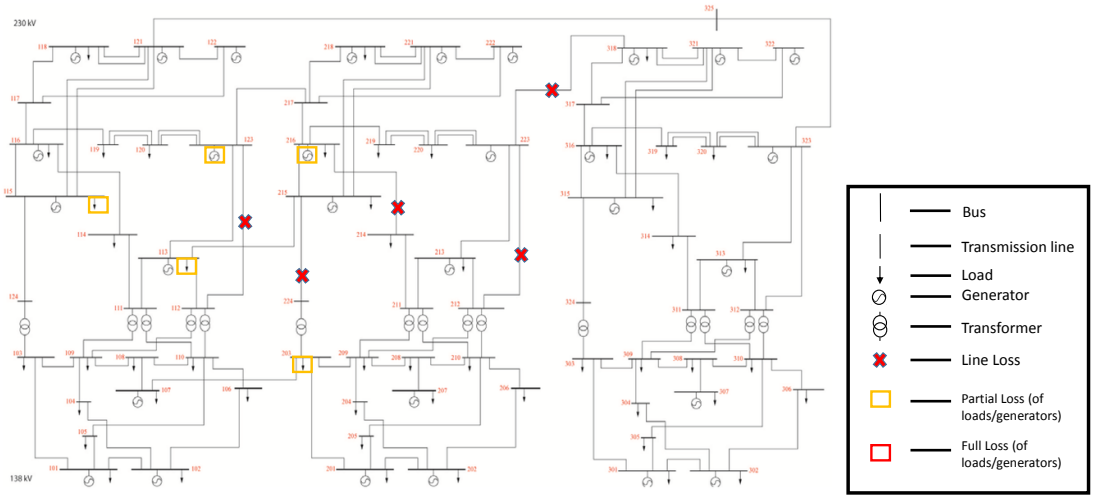

Figure 5.2: Start of cascading failures in the RTS96 case.

After the initial contingency, we can see that other lines begin to trip in effort to protect the transmission lines. As stated earlier, additional components become further stressed. An example of a later stage of the cascading failure is shown in Figure 5.3

At this point we can observe that the network has lost a significant portion of components. Without proper control schemes, the network may be at risk of starting with a small contingency set and ending up in total blackout. Networks are normally designed with redundancy to ensure that the loss of a single component will not have a great impact on the network operation, this is known as $N-1$ security. Further redundancies may be built in a system to improve the robustness of the grid, however may not be economically feasible. Due to this, $N-2$ security is not necessarily ensured for a given

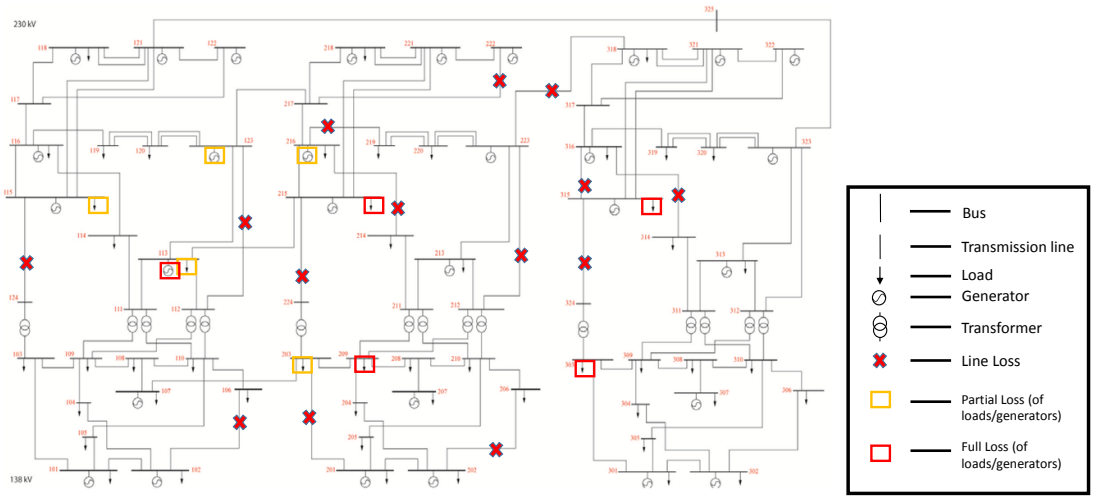

Figure 5.3: Further deterioration of network operation due to cascading.

network. With the difficulty of obtaining this level of security, the ability to recover from said events with intelligent control schemes becomes necessary.

Much work on protective relaying has been performed to mitigate the impacts of contingencies. Most strategies focus on load shedding in attempts to bring voltages level back to tolerable range [11]. Other methods attempt to break the network up into self-sufficient islands to mitigate any propagating failures [26]. This work focuses on a policy based approach making use of actions available to a network operator at given times. Off-line based approaches have yielded success in the past in the form of policy-switching [44]. A policy based approach allows the operator the luxury to perform a sequence of actions that attempt to lead to a common goal such as avoiding blackouts and minimizing component loss.

We focus on an online based approach in the form of policy-rollout [13] which allows one to simulate the outcome of actions on a given model. This approach draws directly upon the work performed in [31]. We perform verification of the work that used the research developed power systems simulator COSMIC [60]. Siemens PSS/e is used as the dynamics simulator of choice in this work.

The dynamics simulator PSS/e is widely used in industry and allows adequate modeling of network components in the dynamic domain. When attempting to model cascading failures, it is important that the simulator has the ability to capture the impacts of device operation with respect to one another in real time. With small step sizes on the order of sub-seconds we can confidently assume that the interaction of components can be adequately modeled.

5.2 Dynamic Simulator

As stated previously, we make use of the commercial dynamic simulator PSS/e. We work with the RTS-96 test case which is comprised of three identical networks connected to one another. Dynamic models are included on the generators in the form of the 'GENSAL' salient generator model, IEEE type 1 exciter and IEEE type 2 governor. The other components, also seen in Figs. 5.1, 5.2, 5.3, include buses, transmission lines, transformers, and loads. For simulating dynamics we implement a time step of $\frac{1}{120}$ seconds specifying how often the network case state is reevaluated.

Basic protection is implemented in the form of overcurrent relays on branches and under voltage/frequency relays on buses. The discrete protective elements are implemented to alleviate local stress within the network and protect components. The addition of this protection scheme will show the impacts of different contingencies and potential cascading failures.

The state of the network is comprised of many different elements that evolve over time. The simulator keeps track of these as state variables which include dynamic variables that change at each time step. The topology of the network is remembered as well which consists of the status of components such as: lines, transformers, loads, and generators. In addition the attached protective relays and their state are saved at each time point as well.

5.3 Policy Rollout

Policy rollout is an attractive solution to network control due to it's ability to be performed online. When tasked with choosing an action, policy rollout may make use of a model and transition function to explore an action set and ultimately choose the best

action to its knowledge. Depending on the model and transition function, an action in a given state may result in a new state with a given probability. This is best represented with equation (5.1) depicting the transition from state s at time t to state s^* at time $t+1$.

$$T(s_t, a_t, s^*_{t+1}) = P(s^*_{t+1}|s_t, a_t) \tag{5.1}$$

One can overcome the issue of probabilistic transitions by running several simulations in a monte-carlo fashion to obtain the average results of performing a particular action. When a model does not have stochasticity, the transition becomes deterministic and eliminates the need to explore a given action at a state multiple times. We make the assumption that our network is deterministic which drastically reduces the time complexity of exploring our action/state space.

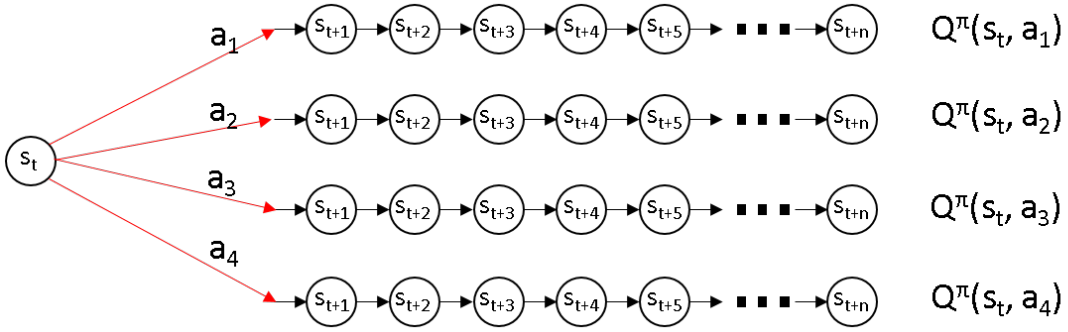

Figure 5.4: Policy rollout with depth one search.

A control policy is implemented based on acting greedily according to an estimated action-value function $\tilde{Q}^\pi$ of a rollout policy π. This action-value function refers to 'how good' an action is at a given state. As shown in Figure 5.4, four actions exist. A depth one search allows each action to be explored with the a baseline policy being implemented afterwards. As an example, we explore each of the four possible actions at s_t, at each time step thereafter we perform the action 'Do nothing.' We can then act greedily by selecting the best action in accordance to equation (5.2). It is important to note that $\tilde{Q}^\pi(s, a)$ is the estimated action-value for a particular action in a unique state. To clarify, the states in Figure 5.4 for each action may be different.

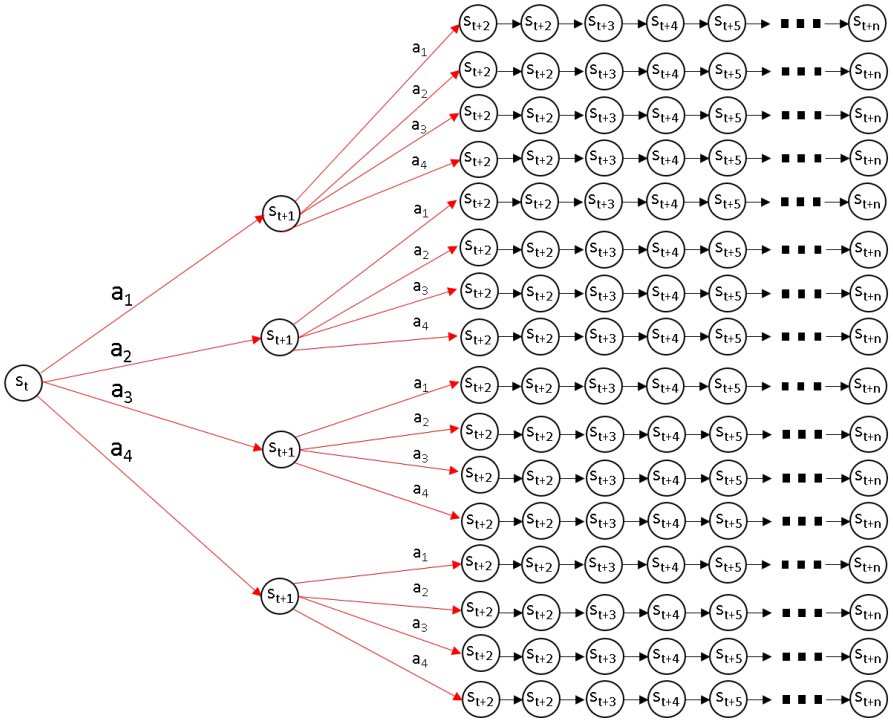

Figure 5.5: Policy rollout with depth two search.

$$\pi_*(s) = \arg\max_{a \in A} \tilde{Q}^{\pi}(s, a) \tag{5.2}$$

The estimated action-value function, $\tilde{Q}^{\pi}(s, a)$, can be found by taking a monte carlo approach. As a result, we would perform the exploration seen in Figure 5.4 many times for each action (necessary if non-deterministic). We will have many potential trajectories based on our action selection, this can be denoted as $\tau = s_0 a_0 \dots s_H$. If we denote the total reward of trajectory τ as $R(\tau)$, it will contain each reward r from each state in the trajectory seen in equation (5.3). Future rewards may be weighted less by using the discount factor $\beta \in [0, 1]$.

$$R(\tau) = \sum_{i=0}^{H} \beta^i r(s_i) \tag{5.3}$$

If we are tasked with estimating $\tilde{Q}^{\pi}(s,a)$, we may create many trajectories and sample m of them [31]:

$$\tilde{Q}^{\pi}_{m}(s,a) = \frac{1}{n_m(a)} \sum_{i=1}^{n_m(a)} R(\tau_i^a) \tag{5.4}$$

In this case, we find the action-value function for state s while taking action a. The total reward of the i^{th} sampled trajectory when taking action a is denoted by $R(\tau_i^a)$. We label the amount of times action a was taken in m samples as $n_m(a)$. After obtaining sufficient samples, we may choose the best action in accordance to equation (5.2).

If time permits, a rollout of larger depth may be performed to further improve upon the implemented policy. For example, a search of depth two would take the form seen in Figure 5.5. This will lead to a drastic increase in exploration complexity, on the order of exponential.

5.4 Application to Network Operation

We use the algorithm policy rollout in tandem with the dynamics simulator PSS/e to demonstrate an improved approach to preventing cascading failures in a network as well as increase load survivability. We focus on the RTS-96 case with a predefined protection scheme. We attempt to improve upon the operation of discrete protective devices and expert based actions.

5.4.1 Baseline Policies

We leverage similar baseline policies shown in [31]. These include: Shedding global load and isolating zones in which contingencies occur. Due to the difficulty with PSS/e interaction of state variables and the necessity of adding user defined models, we did not make use of the 'hysteretic load shed' or HLS baseline policy.

A key thing to note about these policies is that both Isolate and ShedGlobal occur with short delay after the associated contingencies. Similarly, the same delay is implemented with the policy rollout approach to ensure no bias occurrs. The remaining protective elements within the system will continue operating afterwards until the end of simulation. We make the assumption that the ability to shed load and disconnect

certain branches is available which in reality may be limited. In practice, load shedding or branch disconnection is performed by opening a circuit breaker. These devices may not always be located in the necessary configuration, however similar performance should occur.

5.4.2 Available Actions for Policy Rollout

The available actions within our network include both load shedding and islanding. Due to policy rollout being an online method, it is important that the amount of actions is not so great that exploration becomes infeasible. If we allowed load shedding at all available locations concurrently we would come to an action space of the size $O(2^b)$. Due to this we use three expert actions that are also drawn from [31]:

ShedZone(z, p): Shed a proportion of $p \in [0, 1]$ of all loads within zone z.

ShedGloabl(p): Shed a proportion of $p \in [0, 1]$ of all loads within the network.

Island(z): Island zone z from all other zones in the network.

This action space abstraction becomes more necessary as the network scales. Computing power also may impact how an action space is chosen as more power may correspond to the ability to make less abstract actions. Similarly, the ability to search the action space deeper or longer is impacted by available processing power.

5.5 Results

When simulating the RTS-96 case, we do not account for stochasticity. We leverage the baseline policies to get a sense of how well our network can survive certain $N - 2$ contingencies with expert actions. The global load shed action sheds 10% of the load at all shunts and the zone isolation works by disconnecting any tie line connecting a bus to a zone in which a contingency occurs. We also allow no expert action to take place and let only the protection scheme on the network operate. It is important to note that this baseline protection scheme exists for all policies. When testing the policy rollout algorithm we made use of the available actions: ShedZone(z,0.1), ShedGlobal(0.1), and

Island(z) which allows shedding 10% of all shunts in the network, or in a given zone, as well as islanding any zone.

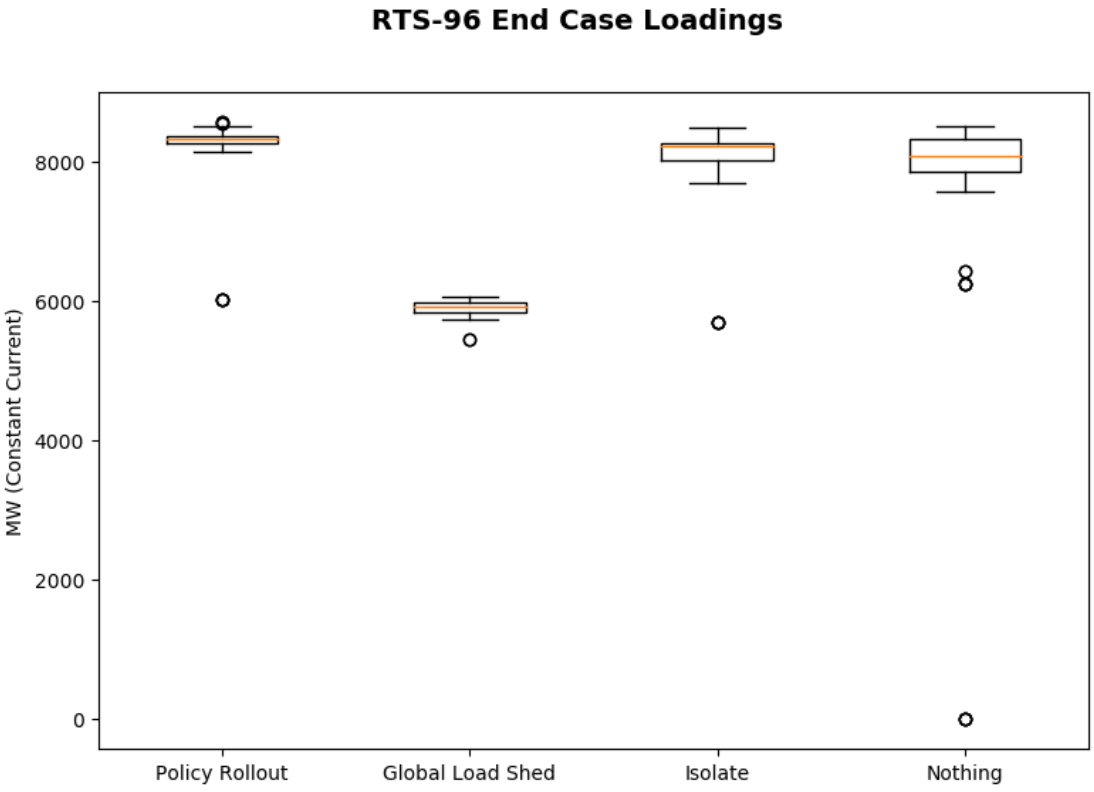

Figure 5.6: RTS-96 end load survivability with different policies.

In Figure 5.7 we see the total survivability of the RTS-96 case. This means we account for the loading at each time point and add up the total amount of load served over the entire duration of the simulation. Conversely, we look only at the end load served in Figure 5.6 to account for how much of the case has survived to the end. Both results from either metric look similar.

An interesting result to observe in Figs. 5.7, 5.6 is that it is possible to perform no expert actions and still obtain good network operation. This relies heavily on how well the protection in the scheme is configured. As seen, the protection scheme allows the case to survive many $N-2$ contingencies. The GlobalShed policy performs worst as it seems to shed unnecessary amounts of load to protect the case. The ShedZone performs

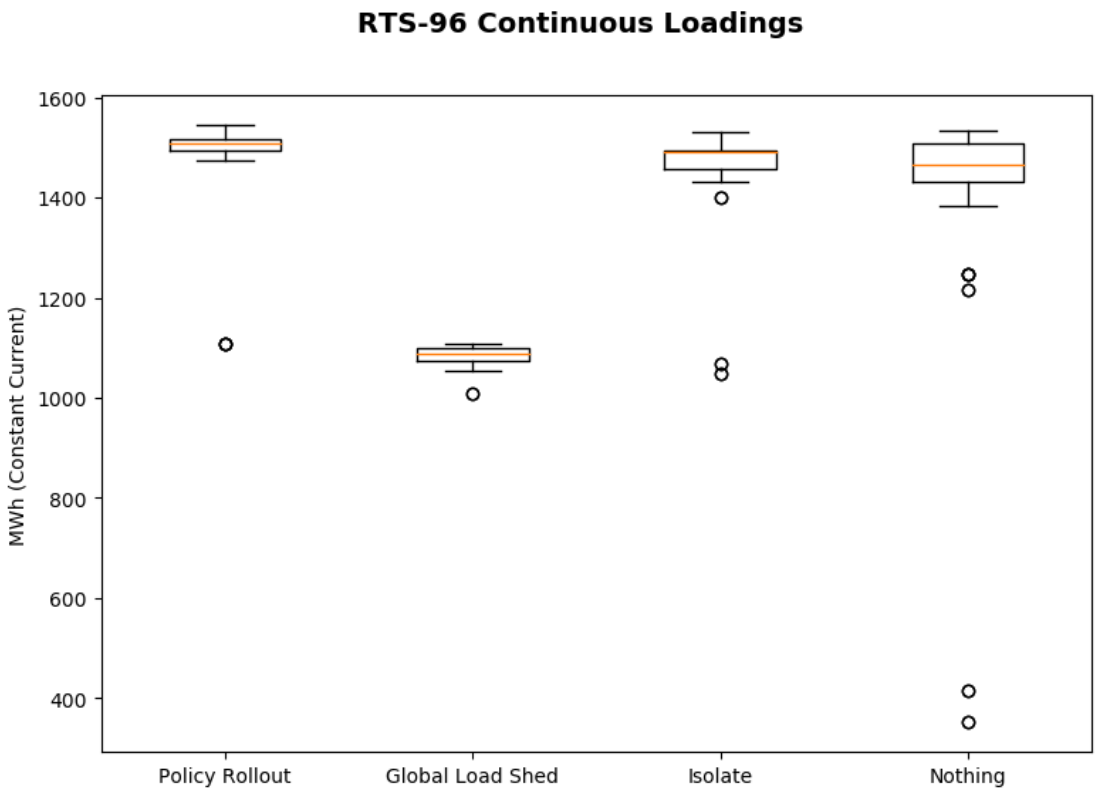

Figure 5.7: RTS-96 total load survivability with different policies.

relatively well, most likely due to the configuration of the RTS-96 case. Depending on the locations of contingencies, it is possible that the baseline protection automatically separates the network into the best islands. The policy rollout case seems to perform the best in which it can allow a higher average surviving load.

With the cases evaluated, it can be seen that the policy rollout approach does seem to perform better that the other potential policies. Further evaluation is necessary for more extreme contingencies. The implementation of stochasticity is also important to check in the future as it will have an impact on the amount of time necessary to evaluate actions.

Chapter 6: Conclusion and Future Work

This work focused on the control of electrical power systems and the impacts and possible mitigation of cyber based attacks. We first discussed the possibility of network damage brought on by a load oscillating smart meter based attack. This attack has proven possible and the intelligent control of said meters was shown to significantly impact not only the network in which the attack originated, but to neighboring locations as well. We then presented a potential framework to control the reconnection of a subnetwork to a main grid. The developed framework leveraged real-time measurements to make predictions of network stability to aid in deciding when to reconnect. The ability to perform with limited measurements increased the robustness of the learning scheme in the face of possible adversarial measurements. The necessity of sub-network control when integrating smaller interconnected networks will be of great importance in the coming years.

The next chapter addressed the need of a pre-processing tool to locate corrupt PMUs when a cyber attack occurs. We made use of the well known nuclear norm minimization technique and discussed another solution in LSTM. The combination of the nuclear norm method and developed learning scheme generated positive results. We also showed the impacts of the lack of pre-processing data on our learning scheme which highlighted the need of said pre-processing. The LSTM based approach was a relatively low time complexity pre-screening method to determine if an attack had occurred. The addition of said method may not be warranted based on the difficulty of scalability. Situations where operators are confident in the current operating point of the grid may make use of this algorithm, however it would require a large amount of data for training that is both attack free and from said operating point.

We finished with a policy rollout technique for determining actions in real time that mitigated network damage in the face of contingencies. It can be shown that policy rollout outperformed all other policies in regards to both total load and amount of load left served. Other interesting things of note include the impact of the underlying protection scheme of a network. It is quite possible that only discrete operations of

protective elements without operator control may yield positive results, however would require well developed discrete protection.

Each portion of this research allows for many avenues for future work. The ability to demonstrate vulnerabilities in an electrical power system allows one to bring to light the impacts of well constructed attacks. The ever growing attack platforms with regards to cyber based communication in the power system needs to be addressed in both forms of prevention and identification. The ability to discern authentic data and malicious data will be extremely important as new control techniques are developed to operate the power grid.

Policy based approaches to determine what actions to take in a network is another important subject to be discussed. Policy rollout is a good start to developing smarter controls, however many trade-offs exist when discussing this technique. Experimenting with more extreme contingencies may show even greater performance difference between rollout and the baseline policies and should be attempted. The possibility of allowing deeper searches or less abstract action spaces by delaying actions may be an interesting topic. Off-line Deep-Q learning may also be a future viable solution for immediate action selection.

Finally the ever growing and changing power grid may make use of control schemes stemming from AI and ML techniques in the near future. These avenues may help in many situations such as identifying fault signatures or aid in network control as seen in Chapter 3. With vast amounts of measurements being made available in the power grid in the future, these powerful techniques may revolutionize how decisions in the grid are made in the coming years.

Bibliography

[1] Estimating the costs and benefits of the smart grid. Technical report, Electric Power Research Institute (EPRI), 01 2011.

[2] Smart meters and smart meter systems: A metering industry perspective. Technical report, Edison Electric Institute and Association of Edison Illuminating Companies, 2011.

[3] Advanced metering infrastructure installations in the U.S.A. `http://www.eia.gov/tools/faqs/faq.cfm?id=108&t=3/`, 2016.

[4] E. Alegria, T. Brown, E. Minear, and R. H. Lasseter. Certs microgrid demonstration with large-scale energy storage and renewable generation. *IEEE Transactions on Smart Grid*, 5(2):937–943, March 2014.

[5] E. Alegria, T. Brown, E. Minear, and R. H. Lasseter. Certs microgrid demonstration with large-scale energy storage and renewable generation. *IEEE Transactions on Smart Grid*, 5(2):937–943, March 2014.

[6] J. Allemong. State estimation fundamentals for successful deployment. In *Power Engineering Society General Meeting, 2005. IEEE*, pages 800–801 Vol. 1, June 2005.

[7] F. Aminifar, M. Fotuhi-Firuzabad, and A. Safdarian. Optimal pmu placement based on probabilistic cost/benefit analysis. *Power Systems, IEEE Transactions on*, 28(1):566–567, Feb 2013.

[8] R. Anderson and S. Fuloria. Who controls the off switch? In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 96–101, Oct 2010.

[9] Adnan Anwar, Abdun Naser Mahmood, and Mark Pickering. Modeling and performance evaluation of stealthy false data injection attacks on smart grid in the presence of corrupted measurements. *Journal of Computer and System Sciences*, 83(1):58 – 72, 2017.

[10] T. M. L. Assis and G. N. Taranto. Automatic reconnection from intentional islanding based on remote sensing of voltage and frequency signals. *IEEE Transactions on Smart Grid*, 3(4):1877–1884, Dec 2012.

[11] Nur Najihah Abu Bakar, Mohammad Yusri Hassan, Mohamad Fani Sulaima, Mohamad Naim Mohd Nasir, and Aziah Khamis. Microgrid and load shedding scheme during islanded mode: A review. *Renewable and Sustainable Energy Reviews*, 71:161 – 169, 2017.

[12] Jason Bell. *Support Vector Machines*, pages 139–160. John Wiley & Sons, Inc, 2014.

[13] D. P. Bertsekas and D. A. Castanon. Rollout algorithms for stochastic scheduling problems. In *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, volume 2, pages 2143–2148 vol.2, Dec 1998.

[14] J. Bialek, E. Ciapessoni, D. Cirio, E. Cotilla-Sanchez, C. Dent, I. Dobson, P. Henneaux, P. Hines, J. Jardim, S. Miller, M. Panteli, M. Papic, A. Pitto, J. Quiros-Tortos, and D. Wu. Benchmarking and validation of cascading failure analysis tools. *IEEE Transactions on Power Systems*, 31(6):4887–4900, Nov 2016.

[15] Jr. Burnett, R.O., M.M. Butts, and P.S. Sterlina. Power system applications for phasor measurement units. *IEEE Computer Applications in Power*, 7(1):8–13, 1994.

[16] T.W. Cease and B. Feldhaus. Real-time monitoring of the TVA power system. *IEEE Computer Applications in Power*, 7(3):47–51, July 1994.

[17] J.E. Chadwick. How a smarter grid could have prevented the 2003 U.S. cascading blackout. In *Power and Energy Conference at Illinois (PECI), 2013 IEEE*, pages 65–71, Feb 2013.

[18] J.E. Chadwick. How a smarter grid could have prevented the 2003 u.s. cascading blackout. In *Power and Energy Conference at Illinois (PECI), 2013 IEEE*, pages 65–71, Feb 2013.

[19] Bo-Juen Chen, Ming-Wei Chang, and Chih-Jen Lin. Load forecasting using support vector machines: a study on eunite competition 2001. *Power Systems, IEEE Transactions on*, 19(4):1821–1830, Nov 2004.

[20] Corinna Cortes and Vladimir Vapnik. Suppor-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[21] E. Cotilla-Sanchez, P. D. H. Hines, C. Barrows, S. Blumsack, and M. Patel. Multi-attribute partitioning of power networks based on electrical distance. *IEEE Transactions on Power Systems*, 28(4):4979–4987, Nov 2013.

[22] E. Cotilla-Sanchez, P.D.H. Hines, C. Barrows, and S. Blumsack. Comparing the topological and electrical structure of the north american electric power infrastructure. *Systems Journal, IEEE*, 6(4):616–626, Dec 2012.

[23] E. Cotilla-Sanchez, P.D.H. Hines, C. Barrows, S. Blumsack, and M. Patel. Multi-attribute partitioning of power networks based on electrical distance. *Power Systems, IEEE Transactions on*, 28(4):4979–4987, Nov 2013.

[24] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos. Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements. *IEEE Transactions on Power Systems*, 31(2):1006–1013, March 2016.

[25] Aurora Geib. How privacy-conscious consumers are fooling, hacking smart meters. Natural News, July 2012.

[26] Mehdi Golari, Neng Fan, and Jianhui Wang. Two-stage stochastic optimal islanding operations under severe multiple contingencies in power grids. *Electric Power Systems Research*, 114:68 – 77, 2014.

[27] Bei Gou. Generalized integer linear programming formulation for optimal pmu placement. *Power Systems, IEEE Transactions on*, 23(3):1099–1104, Aug 2008.

[28] Bei Gou. Optimal placement of pmus by integer linear programming. *Power Systems, IEEE Transactions on*, 23(3):1525–1526, Aug 2008.

[29] N. Hatziargyriou. *Microgrids: Architectures and Control*. Wiley - IEEE. Wiley, 2013.

[30] Kelly Higgin. Smart meter hack shuts off the lights. InformationWeek, October 2014.

[31] Jesse Hostetler. *Monte Carlo Tree Search with Fixed and Adaptive Abstractions*. PhD thesis, Oregon State University, 2017.

[32] K.D. Jones, A. Pal, and J.S. Thorp. Methodology for performing synchrophasor data conditioning and validation. *Power Systems, IEEE Transactions on*, 30(3):1121–1130, May 2015.

[33] S. Kalyani and K.S. Swarup. Classification and assessment of power system security using multiclass svm. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 41(5):753–758, Sept 2011.

[34] S. K. Khadem, M. Basu, and M. F. Conlon. Intelligent islanding and seamless reconnection technique for microgrid with upqc. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 3(2):483–492, June 2015.

[35] C. Lassetter, E. Cotilla-Sanchez, and J. Kim. Load oscillating smart meter attack. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 821–825, Dec 2016.

[36] A. Lawson, M. Goldstein, and C.J. Dent. Bayesian framework for power network planning under uncertainty. *Sustainable Energy, Grids and Networks*, 7:47–57, September 2016.

[37] Chunyan Li, Yuanzhang Sun, and Xiangyi Chen. Analysis of the blackout in europe on november 4, 2006. In *2007 International Power Engineering Conference (IPEC 2007)*, pages 939–944, Dec 2007.

[38] N.W.A. Lidula and A.D. Rajapakse. Microgrids research: A review of experimental microgrids and test systems. *Renewable and Sustainable Energy Reviews*, 15(1):186 – 202, 2011.

[39] S. Liu, X. Feng, D. Kundur, T. Zourntos, and K. L. Butler-Purry. Switched system models for coordinated cyber-physical attack construction and simulation. In *Smart Grid Modeling and Simulation (SGMS), 2011 IEEE First International Workshop on*, pages 49–54, Oct 2011.

[40] S. Liu, S. Mashayekh, D. Kundur, T. Zourntos, and K. L. Butler-Purry. A smart grid vulnerability analysis framework for coordinated variable structure switching attacks. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–6, July 2012.

[41] N.M. Manousakis and G.N. Korres. A weighted least squares algorithm for optimal pmu placement. *Power Systems, IEEE Transactions on*, 28(3):3499–3500, Aug 2013.

[42] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security Privacy*, 7(3):75–77, May 2009.

[43] Stephen McLaughlin, Dmitry Podkuiko, Sergei Miadzvezhanka, Adam Delozier, and Patrick McDaniel. Multi-vendor penetration testing in the advanced metering infrastructure. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 107–116, New York, NY, USA, 2010. ACM.

[44] R. Meier, E. Cotilla-Snchez, and A. Fern. A policy switching approach to consolidating load shedding and islanding protection schemes. In *2014 Power Systems Computation Conference*, pages 1–7, Aug 2014.

[45] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[46] S. Mousavian, J. Valenzuela, and J. Wang. A probabilistic risk mitigation model for cyber-attacks to PMU networks. *IEEE Transactions on Power Systems*, 30(1):156–165, Jan 2015.

[47] Anu Natarayan. *The Emerging Smart Grid: Opportunities for Increased System Reliability and Potential Security Risks*. PhD thesis, Carnegie Mellon University, 2012.

[48] Thomas W. Overton. Oncor's system operating services facility, lancaster, texas. *Power*, 159(8):48 – 51, 2015.

[49] Christopher Parmer, Eduardo Cotilla-Sanchez, Heidi K. Thornquist, and Paul D.H. Hines. Developing a dynamic model of cascading failure for high performance computing using Trilinos. In *Proceedings of the First International Workshop on High Performance Computing, Networking and Analytics for the Power Grid*, HiPCNA-PG '11, pages 25–34. ACM, 2011.

[50] 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*, 12:2825–2830, 2011.

[51] A.G. Phadke. Synchronized phasor measurements in power systems. *Computer Applications in Power, IEEE*, 6(2):10–15, April 1993.

[52] A.G. Phadke. Synchronized phasor measurements-a historical overview. In *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, volume 1, pages 476–479, Oct 2002.

[53] Navigant Research. Number of smart meter installations, worldwide, 2013.

[54] Ibrahim Saeh and M.W.Mustafa. Machine learning classifiers for steady state security evaluation in power system. *International Journal of Computer Science*, 9(2):262–269, 2012.

[55] A. Sankarakrishnan and R. Billinton. Sequential Monte Carlo simulation for composite power system reliability analysis with time varying loads. *IEEE Transactions on Power Systems*, 10(3):1540–1545, 1995.

[56] Elhadi Shakshuki, Khaled Shuaib, Zouheir Trabelsi, Mohammad Abed-Hafez, Ahmed Gaouda, and Mahmoud Alahmad. Resiliency of smart power meters to common security attacks. *Procedia Computer Science*, 52:145 – 152, 2015.

[57] M. A. Sofla and R. King. Control method for multi-microgrid systems in smart grid environmentstability, optimization and smart demand participation. *IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–5, Jan 2012.

[58] M. A. Sofla and R. King. Control method for multi-microgrid systems in smart grid environmentstability, optimization and smart demand participation. *IEEE PES Innovative Smart Grid Technologies (ISGT)*, pages 1–5, Jan 2012.

[59] Saleh Soltan, Dorian Mazauric, and Gil Zussman. Cascading failures in power grids: Analysis and algorithms. In *Proceedings of the 5th International Conference on Future Energy Systems*, e-Energy '14, pages 195–206, New York, NY, USA, 2014. ACM.

[60] J. Song, E. Cotilla-Sanchez, G. Ghanavati, and P. D. H. Hines. Dynamic modeling of cascading failure in power systems. *IEEE Transactions on Power Systems*, 31(3):2085–2095, May 2016.

[61] Jiajia Song. *Dynamic modeling and mitigation of cascading failure in power systems.* PhD thesis, Oregon State University, 2015.

[62] Jiajia Song, Eduardo Cotilla-Sanchez, Goodarz Ghanavati, and Paul D H Hines. Dynamic modeling of cascading failure in power systems. *IEEE Transactions on Power Systems*, 31(3):2085–2095, 2016.

[63] S. Tamronglak, S.H. Horowitz, A.G. Phadke, and J.S. Thorp. Anatomy of power system blackouts: preventive relaying strategies. *Power Delivery, IEEE Transactions on*, 11(2):708–715, Apr 1996.

[64] F. Tang, J. M. Guerrero, J. C. Vasquez, D. Wu, and L. Meng. Distributed active synchronization strategy for microgrid seamless reconnection to the grid under unbalance and harmonic distortion. *IEEE Transactions on Smart Grid*, 6(6):2757–2769, 2015.

[65] Jason Van Hulse, Taghi M. Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 935–942, New York, NY, USA, 2007. ACM.

[66] L. Wehenkel. Machine learning approaches to power-system security assessment. *IEEE Expert*, 12(5):60–72, Sep 1997.

[67] P. Wong, P. Albrecht, R. Allan, R. Billinton, Q. Chen, C. Fong, S. Haddad, W. Li, R. Mukerji, D. Patton, A. Schneider, M. Shahidehpour, and C. Singh. The IEEE

reliability test system-1996. *Power Systems, IEEE Transactions on*, 14(3):1010–1020, Aug 1999.

[68] P. Wong, P. Albrecht, R. Allan, R. Billinton, Q. Chen, C. Fong, S. Haddad, W. Li, R. Mukerji, D. Patton, A. Schneider, M. Shahidehpour, and C. Singh. The ieee reliability test system-1996. a report prepared by the reliability test system task force of the application of probability methods subcommittee. *Power Systems, IEEE Transactions on*, 14(3):1010–1020, Aug 1999.

[69] Jinghe Zhang, G. Welch, G. Bishop, and Zhenyu Huang. Optimal pmu placement evaluation for power system dynamic state estimation. In *Innovative Smart Grid Technologies Conference Europe (ISGT Europe), 2010 IEEE PES*, pages 1–7, Oct 2010.

[70] Yanjun Zhang, Tie Li, Guangyu Na, Guoqing Li, and Yang Li. Optimized extreme learning machine for power system transient stability prediction using synchrophasors. *Mathematical Problems in Engineering*, 2015.

[71] Ming Zhou, V.A. Centeno, J.S. Thorp, and A.G. Phadke. An alternative for including phasor measurements in state estimators. *Power Systems, IEEE Transactions on*, 21(4):1930–1937, Nov 2006.

[72] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas. Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Transactions on Power Systems*, 26(1):12–19, Feb 2011.

APPENDICES

Appendix A: PSS/e Models: Generator Dynamics

A.1 Salient Generator Model (GENSAL)

Table A.1: Salient generator parameters

CONs	#	Value	Description
J			$T'_{do}(>0)$(sec)
J+1			$T''_{do}(>0)$(sec)
J+2			$T''_{qo}(>0)$(sec)
J+3			H, Inertia
J+4			D, Speed damping
J+5			X_d
J+6			X_q
J+7			X'_d
J+8			$X''_d = X''_q$
J+9			X_l
J+10			$S(1.0)$
J+11			$S(1.2)$

Table A.2: Salient generator states

STATEs	#	Value	Description
K			E'_q
K+1			ψkd
K+2			$\psi'' q$
K+3			Δ speed (pu)
K+4			Angle (radians)

A.2 IEEE Type 1 Excitation System (IEEET1)

Table A.3: Excitation system parameters

CONs	#	Value	Description
J			T_R
J+1			K_A
J+2			T_A
J+3			V_{RMAX} or zero
J+4			V_{RMIN}
J+5			K_E or zero
J+6			$T_E(>0)$
J+7			K_F
J+8			$T_F(>0)$
J+9		0	Switch
J+10			E_1
J+11			$S_E(E_1)$
J+12			E_2
J+13			$S_E(E_2)$

Table A.4: Excitation system states

STATEs	#	Value	Description
K			Sensed V_T
K+1			Regulator output, V_R
K+2			Exciter output EFD
K+3			Rate feedback integrator

Table A.5: Excitation system variables

VARs	#	Value	Description
L			Sensed K_E

A.3 IEEE Type 2 Speed Governing Model (IEEEG2)

Table A.6: Governor parameters

CONs	#	Value	Description
J			K
J+1			T_1(sec)
J+2			T_2(sec)
J+3			T_3(>0)(sec)
J+4			P_{MAX}(pu on machine MVA rating)
J+5			P_{MIN}(pu on machine MVA rating)
J+6			T_4(>0)(sec), water starting time

Table A.7: Governor states

STATEs	#	Value	Description
K			First integrator
K+1			Second integrator
K+2			Hydro turbine

Table A.8: Governor variables

VARs	#	Value	Description
L			Reference P_0