Critical Substation Risk Assessment and Mitigation

Jacques Delport

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Electrical Engineering

Virgilio A. Centeno, Chair
James S. Thorp
Arun G. Phadke
Jaime De La Ree
Emanuel E. Bernabeu
Madhav V. Marathe
Amos L. Abbott

April 27th, 2018

Blacksburg, VA

Keywords: critical substations, substation risk, system risk, cascading, hidden failures, misoperations, importance sampling, stability prediction, optimal power flow, game theory, nash equilibria, restricted nash response, counterfactual regret minimization, exploitation, exploitability

Critical Substation Risk Assessment and Mitigation

Jacques Delport

Abstract

Substations are joints in the power system that represent nodes that are vital to stable and reliable operation of the power system. They contrast the rest of the power system in that they are a dense combination of critical components causing all of them to be simultaneously vulnerable to one isolated incident: weather, attack, or other common failure modes. Undoubtedly, the loss of these vital links will have a severe impact to the to the power grid to varying degrees.

This work creates a cascading model based on protection system misoperations to estimate system risk from loss-of-substation events in order to assess each substation's criticality. A continuation power flow method is utilized for estimating voltage collapse during cascades. Transient stability is included through the use of a supervised machine learning algorithm called random forests. These forests allow for fast, robust and accurate prediction of transient stability during loss-of-substation initiated cascades.

Substation risk indices are incorporated into a preventative optimal power flow (OPF) to reduce the risk of critical substations. This risk-based dispatch represents an easily scalable, robust algorithm for reducing risk associated with substation losses. This new dispatch allows operators to operate at a higher cost operating point for short periods in which substations may likely be lost, such as large weather events, likely attacks, etc. and significantly reduce system risk associated with those losses.

System risk is then studied considering the interaction of a power grid utility trying to protect their critical substations under a constrained budget and a potential attacker with insider information on critical substations. This is studied under a zero-sum game theoretic framework in which the utility is trying to confuse the attacker. A model is then developed to analyze how a utility may create a robust strategy of protection that cannot be heavily exploited while taking advantage of any mistakes potential attackers may make.

Critical Substation Risk Assessment and Mitigation

Jacques Delport

General Audience Abstract

Substations are key components to the continued and reliable operation of the power system. Their removal from the power system would severely hinder the system's ability to transport power from power producers to end consumers. As larger weather events and potential threats to the power system are being considered, power system engineers to start considering the impact that losing substations would cause on the system. This work studies the impact on the system associated with losing substation and ranks them to find the most important ones. A probabilistic model is created based on incorrect operations in power system protection elements that historically have exacerbated large events in the power system.

Mitigation of this impact is then studied through two preventative means: changing the operating condition of the current system and adding protection to the substations. This is in order to secure the system before potentially losing the operation of a substation. The operating point change is formulated as a new optimization problem that helps alleviate stress on the system close to the most critical substations found in the earlier model. Protection of these substations is analyzed through game-theoretic means where the utility tries to confuse any potential attackers on which substations actually have true, rigid protection on them. In doing so, on expectation, the damage done to the system may be reduced significantly.

Acknowledgements

I would like to express my gratitude to my advisor, Dr. Virgilio Centeno, for his continuous encouragement and guidance throughout my graduate career. He has been a great pillar of support and has during numerous occasions provided me with invaluable advice for my dissertation as well as in general. I would also like to extend my thanks to Dr. Jaime Del La Ree who was with me since the beginning and was the main motivator behind me pursuing a graduate career. Without him, I truly would be on a different path in life.

I also wish to extend my gratitude to all my committee members for helping to shape and mold this dissertation into what it was. The discussions with Dr. Lynn Abbott, Dr. Madhav Marathe, Dr. Arun Phadke, and Dr. Emanuel Bernabeu have been very enlightening and helped to polish this dissertation. I would like to make special thanks to Dr. James Thorp. He was a constant source of brilliant insight and wonderful discussion, in both power systems and in life. Talks with him inspired many ideas in this work. He will be missed.

I would like to thank all my lab mates, old and new. There are far too many to mention, but the discussions and jokes we shared will be cherished. You have made my time during my graduate career that much better.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 Objective and Overview

Reliability has long been a concern of power systems engineers. In order to make sure electricity is delivered with a measure of certainty and that the power system is secure against disruptive events, many reliability metrics, Loss of Load Probability (LOLP), Loss of Load Expectation (LOLE), Loss of Load Hours (LOLH), etc., have been introduced and deployed. These metrics are well defined and accepted throughout industry and typically involve the N-1 criteria where one element in the system is considered on outage with the goal that the system remain reliable under reasonable outage scenarios. N-1 contingencies have historically allowed power system engineers to consider reliability without having to consider the combinatorial explosion of all possible contingencies and disturbances.

Recently, due to computational availability and worry about larger more threatening disturbances, NERC has increased reliability requirements that go further than the standard N-1 under certain considerations [1]. Under these new considerations and worries, hardiness after disturbances is no longer enough. Included in these requirements is that of the protection of critical infrastructure, whose loss may prevent the system from delivering power reliably. Certain elements in a power system, dependent on many factors, may be more critical to the reliable transport of electricity from power producers to consumers. One of the major critical infrastructure concerns, and one this work focuses on, is power system substations. These substations are nodes in the system that link many other nodes in the system and allow power flow from large distances across the system to different portions of the system. These nodes vary from typical power infrastructure in that they are dense infrastructure points whose loss may severely hamper system operations. Disruption of these substation nodes may even lead to low probability cascading sequences. Even though these cascading sequences may be low probability events, their impact can be quite damaging with large ramifications.

Since 2002 there have been 1060 widespread outages affecting more than 10,000 customers in the United States from a variety of factors [2]. Causes of these outages vary from misoperation, stress relieving actions, protection failures, transient and voltage instability, cascading overloads, attacks, and a variety of others. Weather has been the main initiating culprit causing 679 widespread outages from 2003-2012 [3]. These major disruptions are of interest as their severity to the whole of the economy is quite large. This is historically evident in scenarios such as the blackouts of 1965 and 2003 [4]. In the 2003 blackout, the total

estimated loss due to the blackout is anywhere from four to 10 billion dollars with Canada experiencing a reduced GDP of 0.7% and a loss of 18.9 million work hours [5].

While most causes of large disturbances are not meant to be malevolent, it is not unheard of for malicious entities to attack the power system to cause damage in both a physical and cyber manner. In April 16th, 2013, coordinated snipers attacked a substation on the Metcalf substation damaging 17 transformers and causing millions in damages [6]. In September 29th, 2013, an Arkansas station was infiltrated and set on fire causing more than $4 million in damages [7]. In 2015, a cyber attacker shut down seven 110 kV and 23, 35 kV substations through access through the Supervisory Control and Data Acquisition (SCADA) system blacking out parts of Ukraine [8], and in 2016 a transmission substation was attacked causing loss of approximately one fifth of Kiev's power consumption [9]. Similar substation losses on different power systems may hinder those systems unable to deliver power and potentially cause large economic and physical damage to consumers.

As major weather events have stressed the system and threat of attack has become more of a concern in recent years [3], regulatory action per NERC CIP standards [10] has been taken to make sure that utilities protect their critical substations. It is not feasible to protect all substations from being lost as the number of substations servicing the United States is approximately 50-70,000. However, not all of these are critical and very few of these need to be protected as a small subset of interconnection substations may be truly critical. A recent study from the Federal Energy Regulatory Commission (FERC) has shown that the loss of nine of these substations may blackout the eastern interconnection [11]. However, there are no real standards in place for deciding what critical means. This work aims to find critical substations by the damage that may be done from their loss due to widespread, cascading outages.

In broad terms, substation criticality is a measure of how important a substation is to the performance of the system [12]. A substation is critical if its loss affects the system ability to deliver energy reliably. The greater the impact of the loss-of-substation on the system reliability, the more critical a substation is. There are many ways to determine what metrics to use for this. Does the substation loss affect only local issues or is it system wide? Does the substation provide critical loads? How important is the substation to generation, topology or load? These are simply a few questions that may be considered when trying to rank critical substations and may have less importance to differing systems. Ultimately, however, the expected loss-of-load should weigh heavily on the consideration as the ultimate goal of the power system to constantly provide load with power.

The first aim of this work is to find critical system substations by assessing the risk associated with their loss in the form of expected loss-of-load. The damage caused by the loss of each substation may be considered probabilistic. A loss of a substation in a perfectly protected system will cause immediate loss of load that the substation is serving as well as potential voltage, thermal or stability issues. These issues may or may not exacerbate the disturbance but can be mitigated with schemes and designs. However, human error is known to exist when designing and building protection on the system. Relays, meant to protect system elements against damage, are known to misoperate and exacerbate these types of large disturbance scenarios due to hidden failures within the relay [13]. Protective relays have played an important role in furthering the disturbance in 73.5% of major disturbances [14]. Losing a substation may enable hidden failures in relays and other protection misoperations and large scale cascading blackouts may occur.

A cascading model, with these misoperations modeled as probabilistic, is used to analyze the system impact of losing substations. The cascading model is built within an AC load flow framework to estimate the expected loss-of-load associated with the loss of a substation. Each substation criticality is ranked by their risk to the system. Risk, in general, is defined as probability multiplied by the severity of the event. The probability of cascades is taken into account by producing simple probability distributions of failure on line and generator protection equipment. Severity in this work is defined as the amount of load lost due to the loss of the substation and subsequent cascading. Hence, a substation's risk is defined as the probability of load lost multiplied by the amount of load lost due to unstable islanding, collapse, shedding, etc. The model also incorporates overloading, voltage collapse, transient stability, and sampling to determine varying cascades associated with a substation loss, their likelihood, and their severity.

Voltage stability is estimated with a Continuation Power Flow (CPF) method to give insight into voltage collapse scenarios that may occur and further spread a cascade. Elements to be taken out of service are replaced with equivalent power injections and used as the parameterized loads for CPF. Transient stability is estimated using a supervised, machine learning algorithm, namely random forests. The features (inputs) for these random forests are derived in load flow along with a few snapshots from simplified 2^{nd} order model during these cascades while the labels (outputs) are the true transient stability of each machine in full transient time-domain simulations of the same scenarios. In this way, the random forests learn the connection between these load-flow features and the actual transient stability of each machine and predict when that generator will go unstable in future cascading scenarios. Using this method allows for quicker predictions so that transient stability estimation does not become a severe bottleneck in sampling cascades for risk estimation.

In this way, machine learning is used in a planning sense versus the traditional online prediction methods used. Quick predictions are required so that many may be made throughout the cascading analysis.

After substation risks have been obtained through cascading analysis, the goal becomes to assess and mitigate system risk from the loss of these substations. Risks associated with each line gotten from the cascade model are used to develop a dispatch to mitigate risks associated with each substation. A traditional OPF cost function is augmented with these risks to reduce power flow on these high risk lines to reduce the overall risk of cascading. Regularization is employed to maintain the new dispatch remains close to the traditional OPF dispatch point. Weights are then added to give the new OPF the ability to bias towards least-cost generation or less risk. The new dispatches attempt to reduce the risk of the most critical substations by increasing the cost of dispatching (dispatching closer, more expensive units, redirecting flows, etc.)

Finally, this work examines the interaction between an entity that would like to harm the system by attacking a substation and a utility trying to protect substations against these attacks. Assessing the risk of substations and protecting the truly critical ones is half of the picture. Only protecting these substations may be expensive and would broadcast their criticality to all potential attackers. Instead, it is assumed that the utility has a budget and would like to rigidly defend high risk substations and place fake protection on other substations to fool potential attackers about which substations actually have rigid protection on them. As such, the scenario is analyzed in a three-layer fashion. First, the utility analyzed system risk associated with each substation to get the expected loss-of-load for a substation attack. Next, the utility invests in protection of substations knowing these risks. Finally, the attacker decides which substation to attack seeing the protections added and knowing the system risk associated with each substation. The game is first analyzed with a worst-case attacker with full knowledge of the substation risks under Nash Equilibrium. This may be an attacker with insider knowledge or one who has done similar cascading analysis. The utility's benefit from deviating from the Nash Equilibrium assuming a non-optimal attacker is then analyzed using Counterfactual Regret Minimization. In this way, the utility tries to minimize system risk against a worst-case attacker as well as other non-optimal attackers. The contributions of this work may be summarized as follows:

- Creation of a hidden failure, cascading model incorporating transient instability and voltage collapse to find critical substations through their associated system risk
- Augmentation of a traditional OPF incorporating line risks from cascades to reduce system risk associated with critical substations.
- Analysis of interaction between utility and potential (non)-optimal attackers for substation protection.

1.2 Outline

The rest of the paper is organized as follows. Chapter 2 gives a literature review on work that has been done on each chapter of this work. This includes works done in estimating critical substations, hidden failure models for cascading analysis, machine learning-based transient stability prediction, optimal power flow utilizing risk, and interdiction games in power systems.

Chapter 3 goes over the cascading model. The chapter goes over the basic concepts of the model including the power system fundamental assumptions made within the model. The chapter begins with a look into the probabilistic cascade spreading assumption of hidden failures and misoperations. It then goes on to describe the power system steady-state model, the simulation of this model and the results on the IEEE-118 bus system.

Chapter 4 details the implementation of a Random Forest machine-learning algorithm to predict transient stability within a steady-state cascading framework. It gives a quick overview to decision trees and random forests, and describes the features created within a load-flow framework to try to increase the accuracy of the algorithm. Skewed data is also tackled in this chapter to overcome the issue of class imbalance. Finally, the chapter gives results of the accuracy of the random forests and their application to a hidden failure model.

Chapter 5 modifies a traditional OPF with line risk indices. It introduces the traditional OPF and how the cost function can relatively simply be modified to reduce power flow through high-risk lines. Cascading analysis is then re-analyzed on these new dispatches to find new substation risks. Analysis is then done on weighting of the costs in the new augmented OPF to compare system risk to generation cost.

Chapter 6 analyzes the interaction between a smart, rational attacker that wants to damage the system by taking down a substation and a defender, such as a utility, that wants to protect substations. It introduces a game theoretic model to analyze the interaction, and proves the game is reducible to a smaller, Nash-equivalent game. It then goes over Counterfactual Regret Minimization to analyze the game if the attacker were not completely rational and see how the defender may benefit from the attacker's lack of information or rationality.

2 LITERATURE REVIEW

2.1 Substation Criticality and Risk

Substation reliability assessment has long been an area in literature. More specifically, outages that come from substations have been assessed considering grid reliability to determine transmission/generation reliability [15], [16], substation configuration effects [17], [18], as well as the effect of the protection system on reliability [19]. These tend not to focus on critical rankings of substations but on the impact of substation outages and/or configurations on the power system composite reliability. In general, these are all steady state analysis of reliability that use Markov models, state enumeration, Monte Carlo simulation, etc.

More recently, criticality of the substation as a node in the system has been examined using a Markov model for steady-state analysis and transient stability simulations for generator instability [20]. [21] furthered this model by considering load uncertainty sets. [22] used a network centrality approach from graph theory to rank substations based on the structure of the network, and [23] studied an interdependent network of communication and power using graph theory to find critical nodes under a cascading model. These are higher level models that ignore power system aspects in order to focus more on system topology.

In this work, a cascading model is developed to analyze substation criticality through risk. There has been considerable work done on cascading models that vary from probabilistic to deterministic, high level abstractions to low level models, and that vary in objective, time-scale, operator intervention, etc. The literature on the creation of different types of models is vast, and for an extensive review of models see [24] - [25]. This section is to give a brief review on hidden failure models specifically. When examining power lost versus probability, these hidden failure models have been shown to yield power tails in the probability distribution similar to true disturbance NERC data [26] indicating a simple model that simulates actual large scale disturbances.

[14] introduced a DC hidden failure model for impedance relays with rank one approximations to reduce computational complexity and importance sampling to estimate probabilities. [27], [28] included generator voltage-based hidden failures to a similar model to find weak lines in the system. [29] used an AC load flow with a heuristic random search to find new cascading sequences in lieu of a Monte Carlo method to determine optimal relay upgrades under economic constraints. Exposures of hidden failures that did not initiate during a cascade are taken into account in [30] using a DC model by lowering the a priori probabilities

on each relay given an exposure was seen. [31] uses the model from [30] to study the impact of system loading, hidden failure probability, spinning reserve, and control strategies on expected system risk. These are all models in steady-state analysis that ignore voltage and transient stability in order to decrease the computational burden of simulation.

2.2 Machine Learning for Transient Stability Prediction

The cascading model used in this work implements a transient stability prediction through the use of machine learning. Machine learning methods have been used with good success in power systems for Transient Stability Assessment (TSA) and have the following advantages.

- Faster prediction – Fast prediction of instability is critical as operators need this information quickly to be able to mitigate the effects.
- Generalizability – Many different operating conditions may be included in the training to increase TSA accuracy and robustness over many regions of operation.
- Continual Accuracy improvement – As more data comes in of true system interactions, these methods can only become more accurate towards actually TSA.

The accuracy of these models are highly dependent on the offline training phase and type of data given to the models. If the training data is not indicative of the actual data, then the model will perform poorly.

Most applications of machine learning involve online prediction of stability or predicting the stability margin as this is the area where they tend to shine. The general procedure for these methods is to first collect offline measurements of phasor voltages, rotor angles, etc. to train, validate, and test a classifier, estimator, or some other type of algorithm. During real-time these same types of measurements are fed into the algorithm for a computational inexpensive way of predicting stability or margins. The focus on this review will be supervised learning methods used for transient stability assessment. Machine learning methods for transient stability assessment are mainly constrained to three methods: Decision Trees, SVMs, and Neural Networks.

2.2.1 Decision Tree (DT) Methods

Decision trees are a nonparametric supervised learning method that dominates prediction methods within power systems [32]. They provide a method of prediction of linear and nonlinear patterns through straightforward splitting of the data. They classify data through sequential “if-then” statements based on the

category or range of input variables. A trained tree is simple to understand, can give probability or confidence estimates based on the class probabilities that made it to a particular leaf node, and can have a high degree of accuracy when dealing with nonlinear patterns in data. As such, they have been a powerful tool for predicting transient stability. They do unfortunately have drawbacks such as overfitting and robustness. They can be extremely sensitive to perturbations in the training data. A small change in the data can result in a completely different tree [33].

Within power systems, [34] used decision trees to estimate the critical clearing time of generators. In [35], DTs were tested on the French EHV system to test practical application and problems in applying DTs to real systems. [36] trained decision trees on a post-fault phasor measurements across varying conditions with high accuracy. [37] employed DTs on a small Greek system with high penetration of renewables with a transient prediction accuracy on an independent test set of approximately 95%. [33] made use of an DT ensemble learning method named Random Forests (RFs) along with Fuzzy Logic to reduce the volatility in predicting transient stability with respects to the training data and implement Remedial Action Schemes (RAS) to mitigate cascading scenarios.

2.2.2 Neural Network (NN) Methods

Neural networks (NNs) are an old field of machine learning loosely based on the brain that have been gaining traction again due to higher amounts of data and computation available.. The concept of these networks is that there are neurons that are connected to one another and inputs through weighted connections. These neurons take in input signals based on the weighted sums and “decide to fire” based on the activation function used. The output of these neuron then goes to another layer of neurons or to the output where further activation or output regression/classification is done. As activation functions are chosen to be nonlinear, the combination of weighted sums and activation functions across all neurons make a simple feed-forward neural network with one hidden layer a universal approximator [38]. Given enough neurons, with some mild assumptions on the activation functions used, a neural network can approximate any function. NNs have the drawbacks of requiring large extensive training phases as well as their general lack of interpretability. Once trained, these networks are almost black boxes and can easily overfit training data. However, given enough data, NNs can be very powerful machine learning classifiers.

In power systems, [39] made use of generator kinetic input variables in a feed-forward NN to obtain the estimate the CCT of the system. [40] used energy functions as features to a NN estimate CCT, energy margins, and mode of instability. [41] made use of NN along with statistical information to screen

contingencies based on transient stability to further analyze. In [42], a NN was used with static and dynamic features to rank contingencies in the Hydro Quebec System. [43] applied fuzzy NN on post fault phasor measurement unit on different operating conditions to test prediction robustness of the prediction.

2.2.3 Support Vector Machine (SVM) Methods

Support Vector Machines (SVM) aim to find a way to best separate data in a linear fashion. More specifically, SVMs try to find the hyperplane that creates the largest margin or separation between classes [44]. SVMs have the advantage of having a built in way to incorporate nonlinear kernel mapping, extracting features in high dimensional spaces, and are easy to store as they only require a small portion of the training data once actual built to define the support vectors [44]. Though one of the largest drawbacks is the choice of kernel used to fit the SVM to nonlinear data patterns as kernels are sensitive to over-fitting.

[45] used support vector machines with nonlinear mapping using polynomial and radial basis function kernels for TSA to tackle the curse of dimensionality when the input dimensions become too large in large scale power systems. In [46], dynamic response variables were combined as inputs to predict stability with high accuracy. [47] combined multiple SVMs in an ensemble to combine predictions using multiple time windows to increase perfxormance.

This work attempts to use load flow and 2nd order features combined with simple transient features to predict transient stability in a cascade given a load flow scenario. In this way, the classifier is not being used as a real-time prediction tool but rather, a tool to decrease computational burden of transient stability assessment in the planning sense. Many more cascading cases may be assessed due to the speed of prediction.

2.3 Risk Corrected Optimal Power Flow

There has been a growing sentiment that a deterministic security assessment may not be fully adequate for system reliability. Instead, a push towards more probabilistic or risk-based security assessment is being made. Unfortunately, risk management has wide-ranging meanings in power systems and risk does not have a standardized definition [18]. This is not to say that industry has not been dealing with risk before now. Industry practices involving risk have traditionally involved deterministic practices, e.g. percentage reserve requirements, N-1 contingency analysis, SCED, etc. These have worked well in the past but ignore the probabilistic nature of loads and events on the system.

Traditional Security-Constrained Economic Dispatch (SCED) utilizes an optimal power flow to meet demand given reliability constraints on the system. Usually, this is in the form of thermal line limits and proxies for voltage limits [48]. Contingencies may be explicitly modeled as constraints or through the use of line distribution factors (DFAX) in a SCED program. This analysis increases the reliability of the system but ignores the probability of contingency scenarios. Any constraint that is violated in a traditional SCED problem is treated equally infeasible even if one constraint is more dangerous to the system or more likely. Risk-based dispatch considers also the probabilistic nature of the problem. This risk-based assessment and management has more recently found a welcoming niche in power systems due to renewable and variable penetration making the system ever more uncertain.

Research in system risk management through dispatching tends to follow similar strategies. Define severity for the system, find or assume some prior probabilities for events, and finally minimize or constrain the risk of those events. [49] combines risk constraints with a traditional SCED for post contingency states to have risk below a threshold. Severity is modeled as a piecewise linear function of the flow through a circuit. Probabilities of contingencies are gotten through statistical and historical measures. [50] uses risk as a constraint relaxation on infeasible SCED scenarios. To overcome infeasibilities in the traditional SCED problem, thermal limits are increased while constraining system and contingency risks. In this way, a short-term, feasible solution may be found that does not increase risk. Probabilities are also assumed to be gotten through historical means. [51] uses another piecewise function to define severity while assuming probabilities are known a priori. [52] studied component and composite system risk and introduce AC probabilistic optimal power flow models to study mitigating risk on the system. They study different formulations including controlling risk in constraints and minimizing overall system risk in the objective function.

This work creates a similar optimization problem based on expected loss-of-load from cascades as the risk metric rather than defining a severity function and assuming probabilities. Thus, the goal of the transformed OPF is to minimize system risk with respect to cascading from substation losses.

2.4 Interdiction Games in Power Systems

In general, interdiction games are formulations within a game theoretic framework that model a multi-agent, non-cooperative scenario. In these games, there is a network over which the operator wants to perform or optimize some objective. The disrupting agent(s) or interdictor(s) are agents that want to compromise some part of the network to maximize their own goal while hurting the operator. Interdiction games have been used

widely in networking to assess vulnerabilities of nodes in a network [53] - [54]. These games lend themselves well to bilinear and bi-level formulations, and as long as they remain linear, their optimal solution may be found at the extreme points of the intersection of the two polytopes defining the problem's space [53].

These games are have found a niche within power systems as the power grid is simple a large network. A bi-level combination of interdiction and optimal power flow models have been used to study vulnerabilities in power systems [55], [56]. These problems tend to study some attack of a disrupting agent and the response of the operator to mitigate the effects of said attack. [57] varied the study by examining the problem as a fortification game. That is, the problem is formulated as a tri-level problem where a utility invests in fortification of the network, the agent attacks, and then the operator minimizes the attack through OPF. In [58], line switching was introduced in the bi-level problem as a means of corrective action.

3 CASCADING MODEL

3.1 Protection Misoperations

Hidden failures are failures, which are undetectable during normal operation and occur during other disturbances to the system causing protective relays to potentially exacerbate scenarios unintentionally. While these failures rarely occur, they tend to occur when the system is in a major disturbance. NERC has reported that in 73.5% of major disturbances, protective relays have been involved in some way [14]. Though they are often modeled with relays, hidden failures are applicable to many of the components in the protection system such as current transformers, potential transformers, communication channels, digitizing equipment, etc. These remain undetected in normal operation until some event causes them to malfunction. These failures are of particular importance as their effects tend to appear when the system stressed: during/after faults, low voltages, overloading, or other switching events [59]. There are many hidden failure modes associated with different protection schemes, hardware wiring and logic, and relay coordination [60].

These hidden failures are not the only cause of power system misoperations. According to a 2013 NERC report by the Protection System Misoperation Task Force, 28% of all protection misoperations come from incorrect setting, logic, or design related errors. Approximately half of misoperations in the power system were from relay hardware malfunctions or incorrect settings/logic/design errors in the relay [61], shown in Figure 3-1. Microprocessor relays tend to have many more settings than their electromechanical relay counterparts and account for 76% of all setting/logic/design errors studied in the report [61].

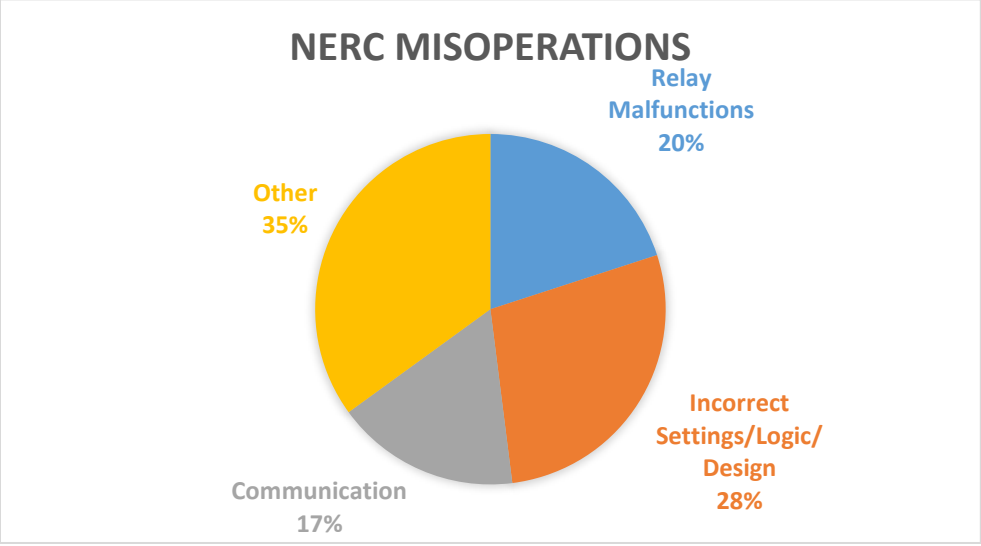

Figure 3-1: NERC misoperations by cause

Misoperations in the protection system are the main probabilistic mechanism of cascading in this work's model. Instead of attempting to model each protection scheme and their associated vulnerability regions [60], simplified probability distributions are used based on apparent impedance of distance relays and voltage ranges of generator protections with exposure regions for each relay.

3.1.1 Impedance-based Misoperation

To estimate expected loss-of-load due to substation outage and thus get a ranking of substation criticality, an AC-based hidden-failure cascading model is used. The model used in this work is similar to the hidden failure models in [14], [27] - [31] with revision and improvements. As in previous models, impedance relays on a line connected to a bus associated with the current event are considered "exposed." This models potential zone two or zone three relays close to the event that may have sympathy trips. All relay probabilities are masked with this binary exposure variable. In this way, only exposed elements have any probability of exposing a hidden failure or enabling misoperations and tripping their associated line. The logic behind this is that historically in cascading scenarios protection misoperations tend to spread in nearby regions and not across the system sporadically. Each exposed relay is modeled as a simple piece-wise probability of tripping based on their seen apparent impedance as shown in Figure 3-2. Note that regardless of hidden failures, a relay will trip with 100% probability if its apparent impedance is too low. Otherwise, it has some small probability of tripping within its zone 3 setting that is meant to model hidden failures within a correctly set relay and an exponentially decaying probability that is meant to model the chance of incorrect settings and miscalibration.

All unexposed relays have binary probabilities of tripping. Their probability is zero if their apparent impedance is greater than their zone one setting (assumed to be 85% of the line in this model) and one elsewise. Elements that were exposed in the current sample cascading path that did not trip have their hidden failure reduced to zero as in [31]. This is due to an exposure being more likely to occur on the first exposure versus future exposures. A revision in this model reinstitutes relay probabilities if the apparent impedances seen by those relays are lower (closer to their zone one settings) in future exposures than in previous exposures. This is meant to model more severe operating conditions potentially exposing failures in the relay that previous exposures did not.

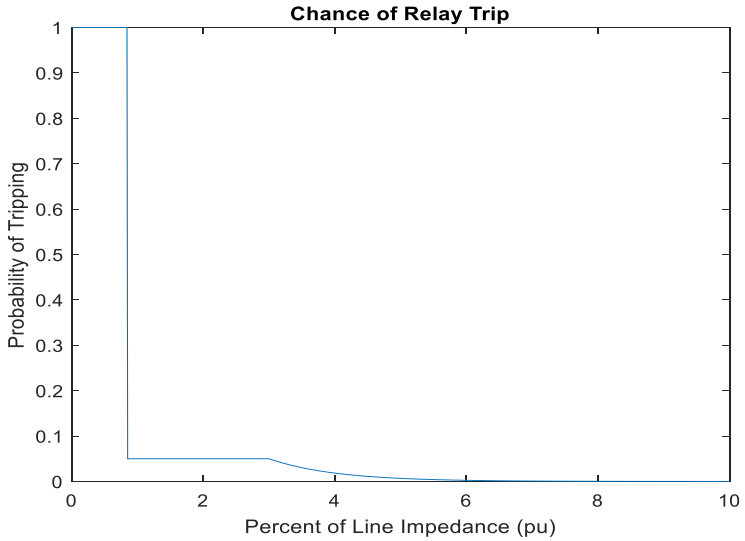

Figure 3-2: Probability of Relay Trip

Traditional hidden failure models assume one-dimensionality to cascades to mimic cascades more realistically [30]. This assumption in other cascading models is based on NERC reports of disturbances [62] and ignores more than one line tripping at a time as it rarely occurs [30]. However, in a large N-k scenario such as loss-of-substation, this may not be the case as many hidden failures may become exposed at once and spread the cascade around the substation that was lost if it spreads at all.

3.1.2 Generator Protection Misoperation

The loss of a generator during a strained system condition further exacerbates the system condition potentially leading to a collapsed system. As such, generator protection misoperations are also considered in this model. It is assumed the generator has zero probability of tripping within a nominal range of terminal voltage and a small constant probability of tripping outside this region as shown in Figure 3-3. These trips are treated the same way as other misoperations in the model in that if a generator trips, all lines connected to its bus are considered exposed as well as the current exposed bus.

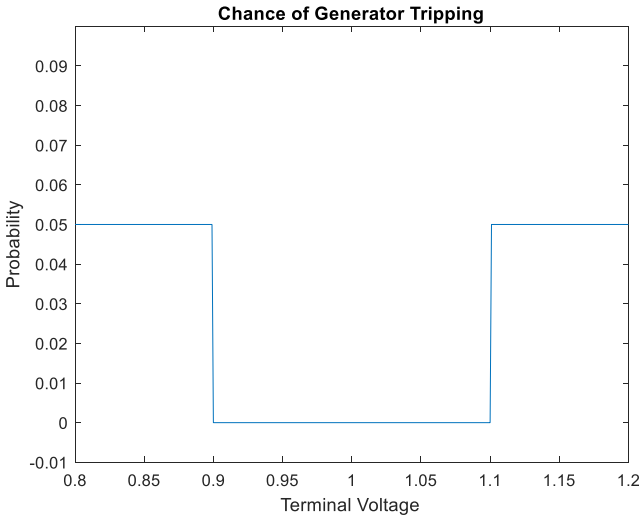

Figure 3-3: Probability of Generator Trip

The non-zero probabilities in Figure 3-3 outside of the nominal range of the generator are meant to model loss-of-excitation protection, over-excitation volts/hertz protection on the GSU transformer, and low voltage protection on the auxiliary equipment of the generator in the plant [63]. The notion behind this is that the generator cannot stay within its reactive capability curve, seen in Figure 3-4, while maintaining nominal voltage.

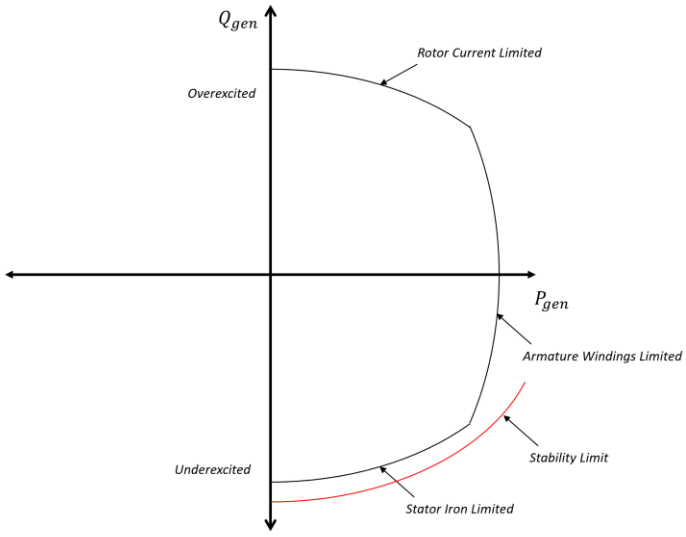

Figure 3-4: Generator Capability Curve

Generators that have lost their excitation can experience serious stator-overloading and extensive power swings [64]. Relays are put in place to trip the generator offline to prevent damage in this scenario. These relays are typically mho-type impedance relays that are meant to trip the machine when the machine becomes too inductive as its field voltage has been lost. If the settings for these relays are not set carefully, the characteristic may encroach under-excited conditions and unnecessarily trip the generator. Volts/Hertz protection is generally applied to generator step up (GSU) transformations-generator pairs to prevent core saturation, large axial eddy currents, and potential breakdown of inter-laminar insulation due to over-excitation in the GSU. This breakdown is due to large flux density saturating the core and causing voltage gradients between laminations [65]. Flux density is directly proportional to voltage and indirectly proportional to frequency. This can be seen by considering the relationship between induced voltage and flux density through a coil, $Ecos(\omega t) = -N\frac{d\phi}{dt}$. Thus volts/hertz relays will protect against over-excitation in the machine.

$$\Phi = -\frac{E}{N\omega}\sin(\omega t) \tag{3-1}$$

The general goal of applying a V/Hz relay is to match the capability curves of both the GSU and the generator to protect against over-excitation while not limiting operating range [65]. Misoperation of both of these types of protection and under-voltage auxiliary equipment protection are modeled through the simplified voltage probability graph.

3.2 Load Flow Model

The model used in this section is a steady-state AC power flow model. Cascading is assumed to spread through overloading of lines, voltage collapse, misoperations in protection systems, and later transient instability. Loads are modeled as constant power, and only traditional generation is considered. Generation is dispatched based on droop control and not an optimal power flow. All loads are considered to have a load-shedding scheme to maintain grid frequency.

3.2.1 Generator Redispatch

Throughout the simulation, load-generation balance has to be maintained. If an island occurs or certain loads and generators are disconnected from the system, the rest of the generators will change their output

power to rebalance the system and normalize the frequency [66]. Unfortunately, load flow is not a convenient tool to model this. Unrealistically, only the slack bus absorbs the change in power required to keep the system balanced due to the assumptions of constant power of the load flow equations. Generators on the grid actually re-dispatch themselves based on primary, secondary and tertiary frequency control. In primary re-dispatch, the governors change the set-point of the generator to stabilize frequency change based on their own control scheme such as droop control. Secondary frequency control changes generator set-points based on automatic governor control (AGC) or economic dispatch while tertiary is based off of some form of area-control [66].

In this work, generators are re-dispatched based on their droop setting modelling primary frequency control only. The thought behind this modeling is that in a large system loss such as this one, there may be quicker cascading scenarios where operators may not be able to react in time or the economic dispatch may not have time to run in the typical five minute time frame [67]. This may not necessarily be the case, but it is thought that this will give a more conservative answer to system risk due to cascading assuming that there are no operational errors made during the cascade. After each event, the load-generation imbalance is recorded as ΔP_L, and the change in system frequency steady-state is calculated from the steady state linear equations as in [66].

$$\Delta w_{ss} = \frac{-\Delta P_L}{\frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n} + D} \tag{3-2}$$

Here, R_i is the droop setting for each generator and D is the system damping assumed to be zero. This is the equation for the change of system frequency of many connected generators based on all of their droop control settings. If Δw_{ss} is too large, load-shedding is implemented as governors cannot react quick enough to stay within a nominal frequency range. Otherwise, generator i changes its power according to equation (3-3).

$$P_i^{new} = P_i^{old} - \frac{\Delta w}{R_i} \tag{3-3}$$

This is called recursively in-case any generators hit their power output constraints and other generators have to increase or decrease their output to compensate.

3.2.2 Load Shedding

In this model, all loads are assumed to have frequency-based shedding and a distributed shedding scheme is enabled. This may be changed to suit the needs of a particular load model. In order to find proper load shedding settings, define p as the average power factor rating of all generators, H as the aggregated system inertia, f_s as the nominal frequency, and L as the relative load excess factor as in [63].

$$L \equiv \frac{\Sigma L_i - \Sigma P_i}{\Sigma P_i} \tag{3-4}$$

The aggregated swing equation for the system may then be re-written as:

$$f\frac{df}{dt} = \frac{pL}{2H}f_s^2 \tag{3-5}$$

We may then integrate this with the condition that at $t_0 = 0, f = f_0$ and get

$$f = f_0\sqrt{1 - \frac{pLf_s^2}{2Hf_0^2}} \tag{3-6}$$

Taking a time-difference average,

$$R = \frac{f_2 - f_1}{t_2 - t_1} \tag{3-7}$$

The average rate of frequency change can then be given as in [63].

$$R = \frac{pL}{H}\frac{(f_2 - f_1)}{\left(1 - \frac{f_2^2}{f_1^2}\right)} \tag{3-8}$$

p: Average power factor rating of all generators
L: Load excess factor
H: Aggregate System intertia
f_1: System frequency right after event
f_2: Future system frequency

The trip time can then be calculated using equation (3-9).

$$t = \frac{f_1 - f_2}{-R} \tag{3-9}$$

Note that these equations assume frequency-independent loads. If the time calculated, plus a tripping delay, is less than the governor response time, distributed load shedding occurs based on the load excess factor. This is to model the situation where load-generation imbalance is so large that the frequency will go too far from nominal before any generator governors can respond and load shedding needs to occur.

3.2.3 Continuation Power Flow

As this model is a full AC model, load-flow convergence is still an issue when power generation/demand balance has been achieved. The solver used in this model is the Newton-Raphson that is well known for diverging when it is not close enough to the new equilibrium [68]. Many scenarios, especially in smaller systems, tend to fail when elements with too much power flowing through them change status as the new equilibrium is not close enough to the current one. This may be overcome by some form of slowly "changing" status. Unfortunately, equilibrium distance is not the only reason NR fails. Newton-Raphson requires that the load-flow Jacobian be non-singular to find a new solution. As cascades spread, voltage tends to collapse due to lack of reactive support in an area, reactive losses on overloaded lines and a variety of other reasons. This voltage collapse, whether it is one bus or the entire system, will cause the Jacobian to become singular and the solver will fail. In the real system, this is not indicative of what would happen as the whole system may not necessarily fail. As voltage on a bus collapses, impedance relays on lines connected to that bus would start seeing abnormally low impedances until the apparent impedances eventually encroaches on their zones of protection, tripping out the lines.

To alleviate these problems, Continuation Power Flow (CPF) was utilized. CPF is a continuation tool that traditionally tries to trace out the PV curve at a bus for a loading set [69] as in Figure 3-5.

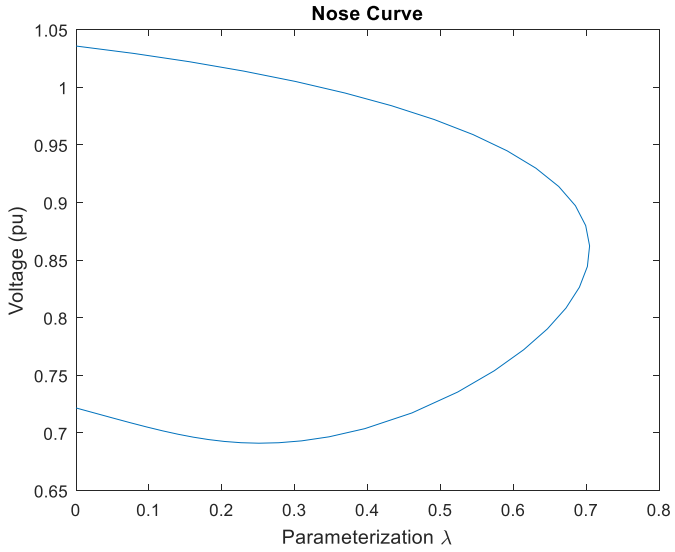

Figure 3-5: PV Curve

It starts at a base operating point and traces the curve through what is known as a predictor-corrector algorithm. The most common approach to tracing the curve is through a tangential approach. The derivative of the PV curve at the current operating point is taken. This tangent is then used to predict the next operating point as loads are increased. Augmented, nonlinear power flow equations are then used to find the actual solution close to the operating point. This method has the benefit of finding the critical point in a power system without difficulty [69]. Loading on the system is parameterized through a variable, λ. The power flow equations are then augmented to obtain $F(\delta, V, \lambda) = 0$ or expanded out.

$$P_{G_i}(\lambda) - P_{D_i}(\lambda) - P_{inj_i} = 0 \tag{3-10}$$

$$Q_{G_i} - Q_{D_i}(\lambda) - Q_{inj_i} = 0 \tag{3-11}$$

Here, P_{inj_i} and Q_{inj_i} are the traditional power flow injection equations.

$$P_{inj_i} = \sum_{j=1}^{N} V_i V_j y_{ij} \cos(\delta_i - \delta_j - \theta_{ij}) \tag{3-12}$$

$$Q_{inj_i} = \sum_{j=1}^{N} V_i V_j y_{ij} \sin(\delta_i - \delta_j - \theta_{ij}) \tag{3-13}$$

$P_{G_i}, P_{D_i}, Q_{D_i}$ are augmented such that they are a function of their base values and the parameter λ.

$$P_{G_i}(\lambda) = P_{G_i}^0(1+\lambda) \tag{3-14}$$

$$P_{D_i}(\lambda) = P_{D_i}^0(1+\lambda) \tag{3-15}$$

$$Q_{D_i}(\lambda) = Q_{D_i}^0(1+\lambda) \tag{3-16}$$

Note here that Q_{G_i} does not get augmented as it is purely dependent on the actual power flow equations. The first step in the predictor-corrector step is the actual prediction. To do this, the Jacobian matrix that gives partial derivatives is augmented by one column.

$$F_\partial = [J_{LF}\ F_\lambda] = [F_\delta\ \ F_v\ \ F_\lambda] \tag{3-17}$$

To obtain the tangent vector, we find the orthogonal vector to the partial derivatives.

$$[F_\delta\ \ F_v\ \ F_\lambda]\begin{bmatrix} d\delta \\ dV \\ d\lambda \end{bmatrix} = 0 \tag{3-18}$$

Here, $T = [d\delta\ \ dV\ \ d\lambda]^T$ is the tangent vector we wish to find. The tangent vector is normalized by a zero vector with a ± 1 only in the k[th] location as the prior equation has one more unknown that equations. This guarantees a nonzero norm on the tangent vector, T, so that the augmented system is nonsingular at the nose of the curve. The new system can be solved for T to find the tangent vector.

$$T = \begin{bmatrix} F_\delta\ \ F_v\ \ F_\lambda \\ e_k \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} J_{LF} \quad F_\lambda \\ e_k \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{3-19}$$

The prediction step is then

$$\begin{bmatrix} \delta^{(pred)} \\ V^{(pred)} \\ \lambda^{(pred)} \end{bmatrix} = \begin{bmatrix} \delta^{(old)} \\ V^{(old)} \\ \lambda^{(old)} \end{bmatrix} + \sigma T \tag{3-20}$$

These new values are then corrected through solving the nonlinear system.

$$F(\delta, V, \lambda) = 0 \tag{3-21}$$

$$\lambda - \lambda^{pred} = 0 \tag{3-22}$$

Instead of ramping up system loading, this work utilizes CPF as a tool to switch out elements and see which buses, if any, will have voltage problems as the line goes out. Since CPF varies system loading and generation, elements to be switched out are replaced by equivalent injections. The base CPF case is then set with these elements at their full injection, and the target case for the CPF has these elements with zero injection. Hence, the injection reducing to zero represents the element "slowly being taken out of service". As CPF can show what happens to voltage even at singularity, buses that reach their nose-point or singularity (buses whose voltage will go unstable) before the element is completely taken out are to be taken out. The algorithm for CPF with bus collapse is as follows:

1. Convert elements to be removed by equivalent injection.
2. Perform CPF until injection is zero.
3. Find buses that have reach their nose curve before CPF finished.
4. Disconnect those buses. Branches attached to bus are elements to be removed in next CPF.
5. Go to 1.

As such, this algorithm will go through and continuously trip out elements due to voltage collapse until voltage stabilizes or the system has collapsed.

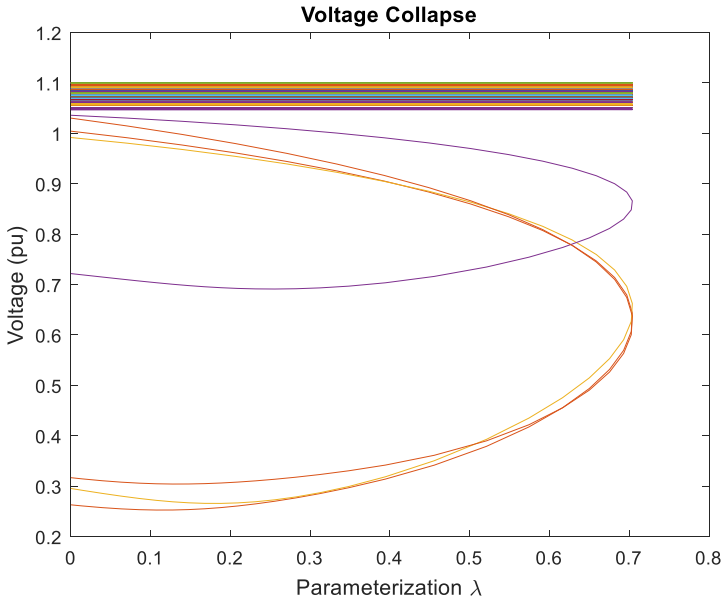

Figure 3-6: Voltage collapse after loss of substation

3.3 Importance Sampling

When considering possible cascading scenarios, a combinatorial problem arises. The amount of all possible cascades to consider grows exponentially with system size. A system with N elements that may trip out has 2^N different possible configurations. Granted, not all these configurations should be considered in cascading scenarios as a cascade is a growing, evolving phenomenon and not one that randomly takes out elements throughout the entire system, but even with the assumption that the cascade remains close by, the amount of cases to consider is simply too large to enumerate through. Hence, the true risk to the system of substation loss places too much computational burden to find. Traditionally to estimate this risk, Monte Carlo simulations are used where states are randomly sampled based on their probabilities [70].

The probabilities of a cascading simulation are very low and would require a Monte Carlo simulation many iterations to estimate the true probability to the point of intractability. The deeper cascading trees would rarely be sampled with cascades often not occurring. The issue is that as hidden failures have low probability of becoming active, deeper cascades would be in the tails of a very peaky distribution. A traditional Monte Carlo estimator would gather most of its samples where very few hidden failures are exposed resulting in poor estimates for the tails of the distribution where the deeper, more severe cascades reside.

One method to get around this is to use importance sampling [14]. Importance sampling is a Monte Carlo-based variance reduction technique that uses a biased distribution instead of the true probability distribution to estimate the true distribution [70]. The system states are sampled from the biased distribution and reweighted to account for the use of a biased distribution to obtain an unbiased estimate [70]. Namely, the probabilities of rare cascades are altered to occur more frequently and accounted for when estimating the true probability. In this way, we obtain more samples from deeper cascades to give better estimates. More formally, given an original, nominal distribution $p(x)$; a new, importance distribution $q(x)$; and a function $f(x)$, an importance sampling estimate of $\mu = E_p[f(x)]$ is:

$$\hat{\mu} = \frac{1}{N}\sum_{k=1}^{N}\frac{f(X_i)p(X_i)}{q(X_i)}, \qquad X_i \sim q(X) \tag{3.23}$$

As an example, consider the PDF in Figure 3-7. The original distribution is very peaky and would require many iterations of sampling to estimate the tails with any accuracy if using the original distribution. Instead, a new a distribution is sampled that gives more importance to the tails of the distribution. This new distribution should have wider tails than the original distribution to lend larger importance to deeper cascades,

have similar shape to the original distribution to avoid increasing variance in the estimation, and its support should include the support of the original distribution (i.e. it should never be zero when the original distribution is nonzero) [70].

The unbiased estimate is given for a given sample cascading path i in equation (3.24).

$$\hat{\rho}_i = \frac{1}{N}\sum_{k=1}^{N}\frac{p_{i,k}^{actual}}{p_{i,k}^{simulated}} \tag{3.24}$$

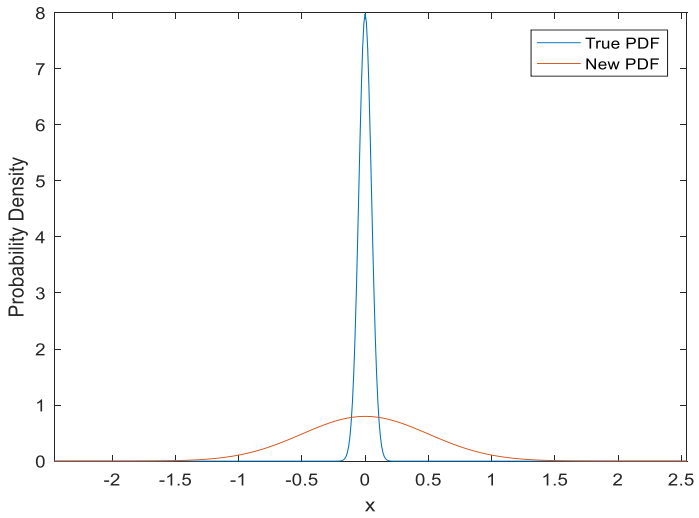

Figure 3-7: Importance Sampling Distributions

Here N is the total number of samples drawn and $p_{i,k}^{simulated}$ is the altered probability of the sample path at iteration k for sample i. [14] found that the main sample cascading paths varied significantly depending on the biased distribution used. Instead of mapping probabilities with a constant mapping, they instead used a probabilistic variation. The same variation on importance sampling is used in this work. At each iteration, the probabilities of each line are altered using a variation where we draw on many different types of distributions

$$p_{i,k}^{simulated} = 0.5\left(\frac{p_{i,k}^{actual}}{p_{i,k}^{max}}\right)^{\mu_k} \tag{3.25}$$

where μ_k is a uniform random variable between [0, 1]. This allows sample draws during any given iteration from many different distributions ranging from a similar variance-scaled distribution, a uniform-distribution, and anything in-between. Doing this allows a great variety in the number of cascades that are sampled. The importance sampling method is combined with risk (probability multiplied by severity) to determine the

stopping criteria for a particular cascade. The sampling technique by itself will stop sampling a cascade when no hidden failures/misoperations are exposed. However, there may be cascade scenarios that go beyond cascades of interest. If the probability of a large cascade is low enough and the severity of such a cascade is also low enough, the cascade is not of interest and is not considered. Doing this allows us to further reduce the computational time required for the simulation.

As an example, a system small enough to enumerate has been created in Figure 3-8 after line Z has tripped. The probabilities of hidden failure have been set constant for simplification purposes. It is a hard task to visualize if the new sampling distribution is similar to the true distribution as it is a high dimensional function dependent on all element statuses. However, we may get an intuition for how similar they are by instead doing a mapping from the original distribution to a new one that is only dependent on the number of outages in a cascade. This new distribution thus gives the probability that a certain number of elements are on outage due to the cascade. The same exposure logic has been applied as stated in 3.1.1, and the states have been enumerated as shown in

Table 3-1. At first only lines A, B, or C are exposed and may trip, and lines D and E only become exposed if any of those three trip.

The second most probable event has a probability of 9.99e-05 and would take approximately 10,000 samples to see once on expectation. Importance sampling allows us to visit this state much more often. Figure 3-7 shows the comparison of the true distribution of number of outages from a state enumeration method, the estimated distribution from importance sampling, and the average scaled importance sampling distribution after 10,000 samples.

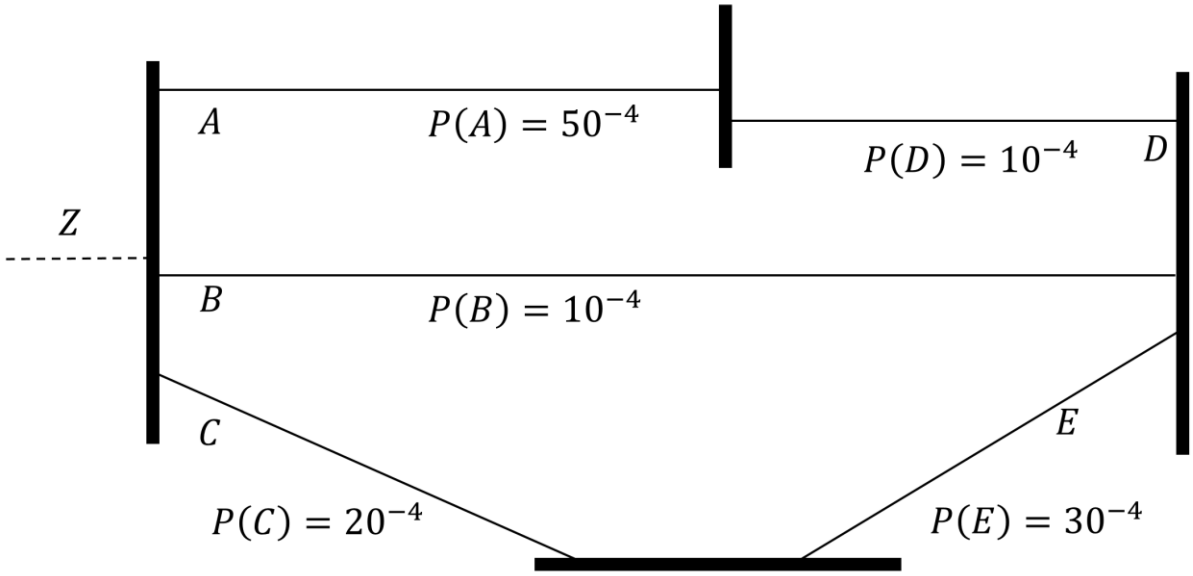

Figure 3-8: Small example system

Table 3-1: Enumerated states of small system

Layer 1	Layer 2	Probabilities	Probability
A	NONE	A*(1-B)*(1-C)*(1-D)	0.000499800024999
	D	A*(1-B)*(1-C)*(D)	9.9980001e-08
B	NONE	(1-A)*(B)*(1-C)*(1-D)*(1-E)	9.99100249973001e-05
	D	(1-A)*(B)*(1-C)*(D)*(1-E)	1.99860021999e-08
	E	(1-A)*(B)*(1-C)*(1-D)*(E)	9.9920016999e-09
	DE	(1-A)*(B)*(1-C)*(D)*(E)	1.9988001e-12
C	NONE	(1-A)*(1-B)*(C)*(1-E)	9.99300109995e-05
	E	(1-A)*(1-B)*(C)*(E)	9.9940005e-09
AB	NONE	(A)*(B)*(1-C)*(1-D)*(1-E)	4.99800024999e-08
	D	(A)*(B)*(1-C)*(D)*(1-E)	9.9980001e-12
	E	(A)*(B)*(1-C)*(1-D)*(E)	4.9985001e-12
	DE	(A)*(B)*(1-C)*(D)*(E)	9.999e-16
AC	NONE	(A)*(1-B)*(C)*(1-D)*(1-E)	4.99800024999e-08
	D	(A)*(1-B)*(C)*(D)*(1-E)	9.9980001e-12
	E	(A)*(1-B)*(C)*(1-D)*(E)	4.9985001e-12
	DE	(A)*(1-B)*(C)*(D)*(E)	9.999e-16
BC	NONE	(1-A)*(B)*(C)*(1-D)*(1-E)	9.9920016999e-09
	D	(1-A)*(B)*(C)*(D)*(1-E)	1.9988001e-12
	E	(1-A)*(B)*(C)*(1-D)*(E)	9.993001e-13
	DE	(1-A)*(B)*(C)*(D)*(E)	1.999e-16
ABC	NONE	(A)*(B)*(C)*(1-D)*(1-E)	4.9985001e-12
	D	(A)*(B)*(C)*(D)*(1-E)	9.999e-16
	E	(A)*(B)*(C)*(1-D)*(E)	4.99900000000000e-16
	DE	(A)*(B)*(C)*(D)*(E)	1.00000000000000e-19
None	NONE	(1-A)*(1-B)*(1-C)	0.999300109995000

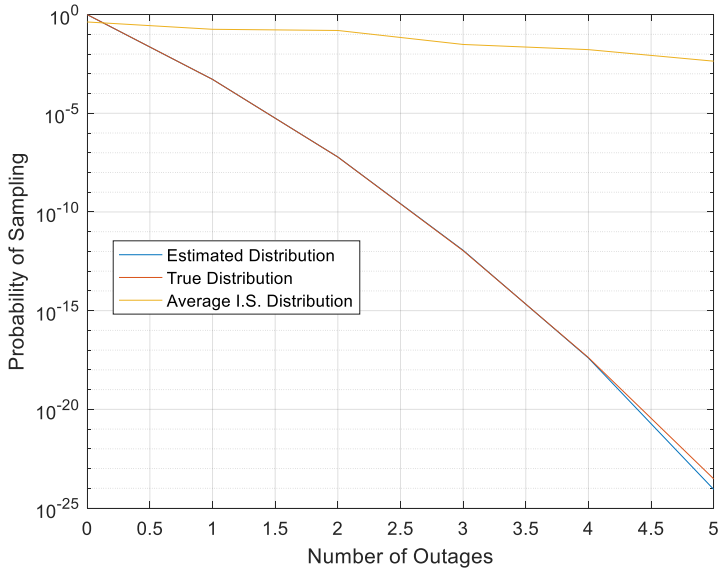

Figure 3-9: Distribution Comparisons

The estimated distribution from importance sampling is nearly identical to the true distribution using 10,000 samples. As the importance sampling distribution changes from sample to sample, the mean distribution is given instead of each unique distribution. Importance sampling lowers the probability of no cascade happening (zero elements on outage) and increases the probability of the largest cascades happening. A variation is used in this work where importance sampling for sampling the cascading paths, but the true system state probabilities are recorded. Given enough samples to produce all unique, significant sample paths, the sum of the probabilities will be a tight lower bound to the probability of cascading. The Monte Carlo simulation is stopped when the sample variation of the number of unique of distinct cascading samples drops too low.

The flowchart for the cascading model algorithm is shown in Figure 3-10. Once the substation is lost, true operations are checked (instability, overloads, etc.) and tripped out. Exposed elements are updated with respect to elements that have already tripped out of service and their exposure regions. Once no more true operations occur, the risk of the current cascade is checked to limit the number of cascades to check. If a cascade is of particularly low probability and severity, there is no need to further consider it spreading. Assuming a cascade is of high enough risk, misoperations are then sampled based on the exposed elements. This cycle repeats until either the risk is low enough or there are no misoperations sampled. At this point, the cascade is considered ended, and its properties are recorded. The simulation is then reset and new cascades are found until the number of unique cascades has seemingly converged. The metric for convergence in this work is no new cascades have been found within the previous 10^4 sampled cascades.

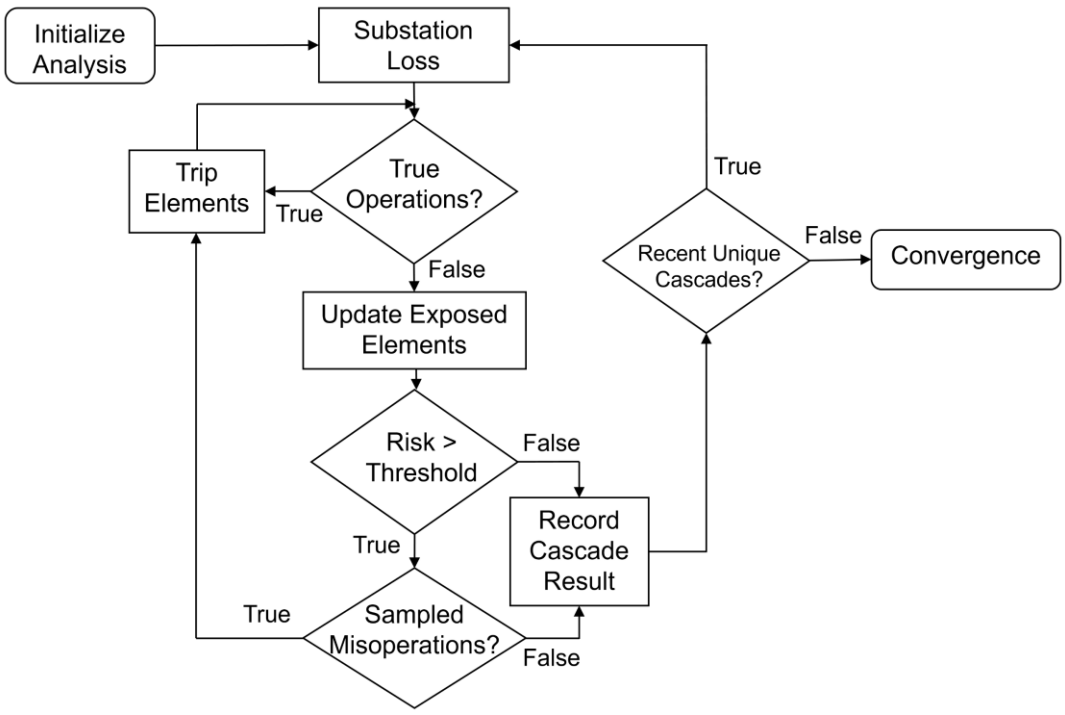

Figure 3-10: Model Flowchart

3.4 System Impact

Let q_m^i be a misoperation cascading sequence, *m*, after the loss of substation *i*. We then define β^i to be the set of all M possible unique blackouts associated with substation *i* and β to be the set over all *Z* substations of these sets.

$$\beta^i = \{q_m^i \ \forall\, m = 1, \dots, M\} \tag{3-26}$$

$$\beta = \{\beta^i \ \forall\, i = 1, \dots, Z\} \tag{3-27}$$

The expected loss-of-load, P_l, if substation *i* has been lost can then be defined as

$$E[P_l \mid i\,] = \sum_{\beta^i} P_m^i \rho^i{}_m \tag{3-28}$$

where P_m^i and ρ_m^i is the load lost and the probability associated with a unique sample cascade q_m^i, respectively. Assuming uniform distribution of beliefs of losing any substation, the overall expected loss-of-load of the system can be calculated by assuming the probability of a cascade is independent from the

probability of losing a substation. In this manner, both the risk associated with a particular substation and the overall system risk may be obtained from the cascading analysis.

$$E[P_l] = \frac{1}{Z}\sum_{i=1}^{Z} E[P_l \mid i\,] = \frac{1}{Z}\sum_{i=1}^{Z}\sum_{k=1}^{M} P_k^i \rho^i{}_k \tag{3-29}$$

These results can also be used to obtain the risk associated with each line in the system. Lines that misoperate during higher risk cascading sequences contribute to the overall risk of the system. If the line risks are summed across all cascading sequences, they made be ordered from high to low risk lines. Set C_r to be the subset of all possible unique cascading sequences that contain line relay *r*.

$$C_r = \left\{ q_m^i \middle| r \in q_m^i \;\; \forall \;\; i \in \{1, \dots Z\},\; m \in \{1, \dots, M\} \right\} \tag{3-30}$$

We may then define the expected loss-of-load associated with any line relay as

$$E[P_l^r] = \sum_{m \in C_r} P_m \rho_m = \frac{1}{Z}\sum_{i=1}^{Z}\sum_{m \in \beta^i} P_m \rho_m = \frac{1}{Z}\sum_{i=1}^{Z}\sum_{m=1}^{M} I_{C_r}\left(q_m^i\right) P_m^i \rho^i{}_m \tag{3-31}$$

where $I_{C_r}(\cdot)$ is the indicator function defined as:

$$I_{Cr}(\omega) = \begin{cases} 1 & \text{if } \omega \in C_r \\ 0 & \text{otherwise} \end{cases} \tag{3-32}$$

Note that since q_m^i is a misoperation sequence, it only contains lines that were tripped by a misoperation not those tripped correctly due to voltage collapse or thermal overload. This allows us a measure of how much a misoperation in a relay is costing the system to help prioritize relay upgrades and maintenance for maximizing reliability under a constrained budget.

3.5 Simulation Results

The simulation was performed on a modified IEEE 118-bus system with loading changed and generators redispatched. The new operating point is shown in Appendix A – Base Operating Point. Matpower [71] was used as the load-flow program for all Newton-Raphson calculations as well as the modified base OPF. A substation in this study was defined as any buses connected by a transformer or zero impedance line. Buses without transformers were considered their own substation. A typical N for each substation study was

on the order of 10^5-10^6 but was given a bare minimum of 10^4 of samples. *p* for both hidden line failures and hidden voltage failures was set to 0.05. Convergence was considered when no new distinct cascades were found in the previous 10^4 samples. For each substation, the recorded expected loss-of-load is shown in Figure 3-11 for both the AC model and its DC counterpart. The DC counterpart underestimates expected loss-of-load as it does not include voltage. The optimistic estimate on relay apparent impedance in the DC model leads to this bias. However, the critical ranking of the substations does not to change significantly.

Figure 3-12 shows the expected loss-of-load due to cascading. The nuance here is that Figure 3-11 includes any damage done during the actual loss-of-substation without hidden failures. Here, the ranking can change quite drastically. One interesting thing to note is there are two substations that the DC model is over-estimating compared to the AC model when considering cascading risk (substations 50 and 57). During the cascading analysis, lines connected to another bus started seeing low voltages. These voltages eventually collapsed bus isolation from the system. This collapse and isolation led to change in line flows near the exposed elements in the AC model leading to lower overall expected loss-of-load.

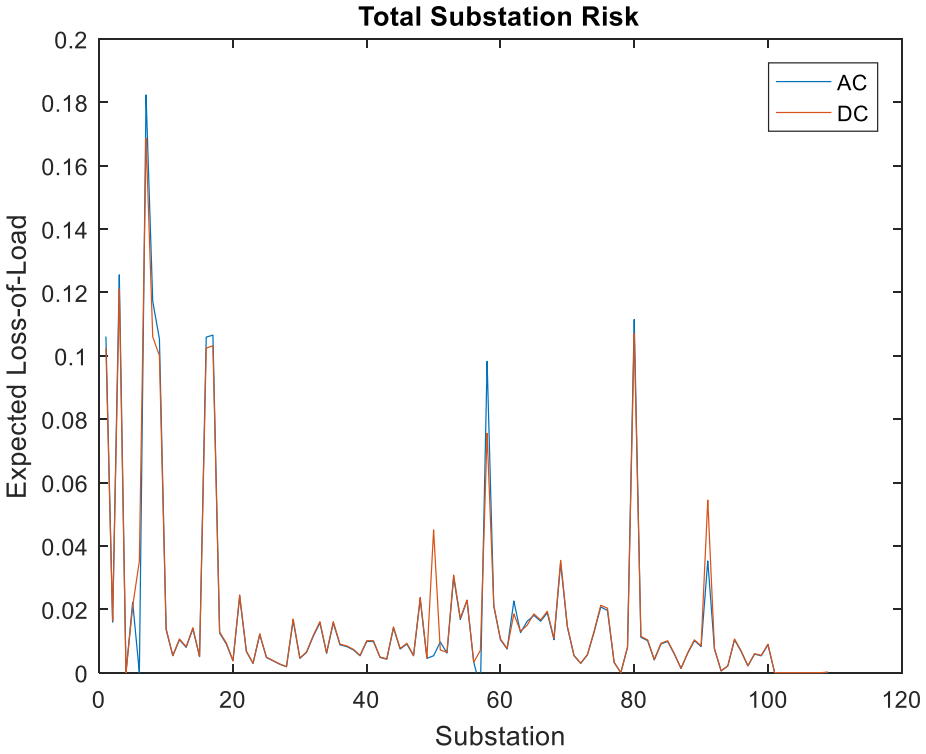

Figure 3-11: Expected load lost under substation failure

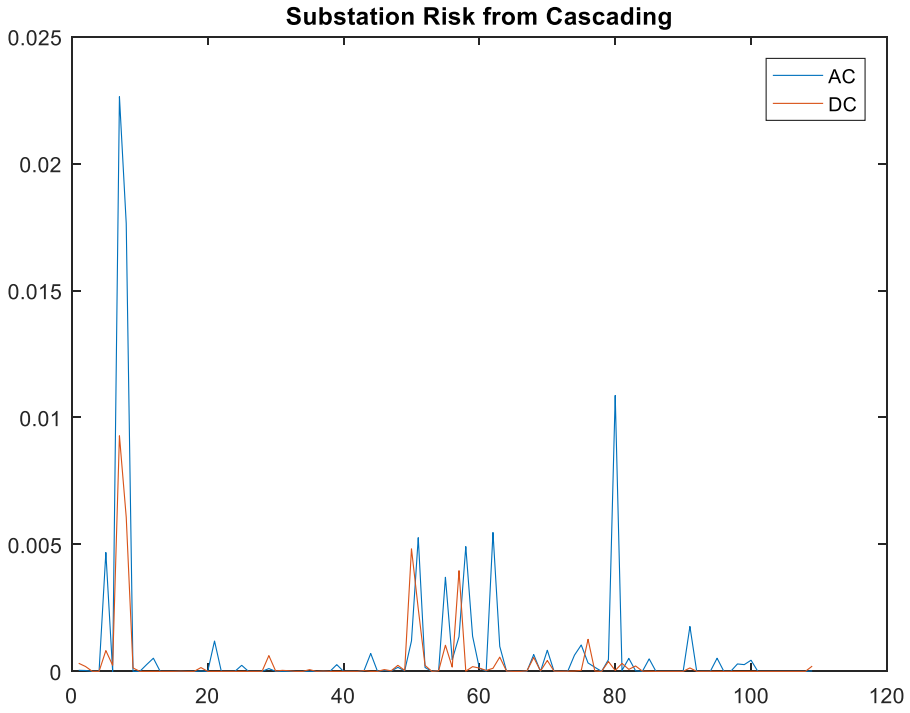

Figure 3-12: Expected load lost due to cascades

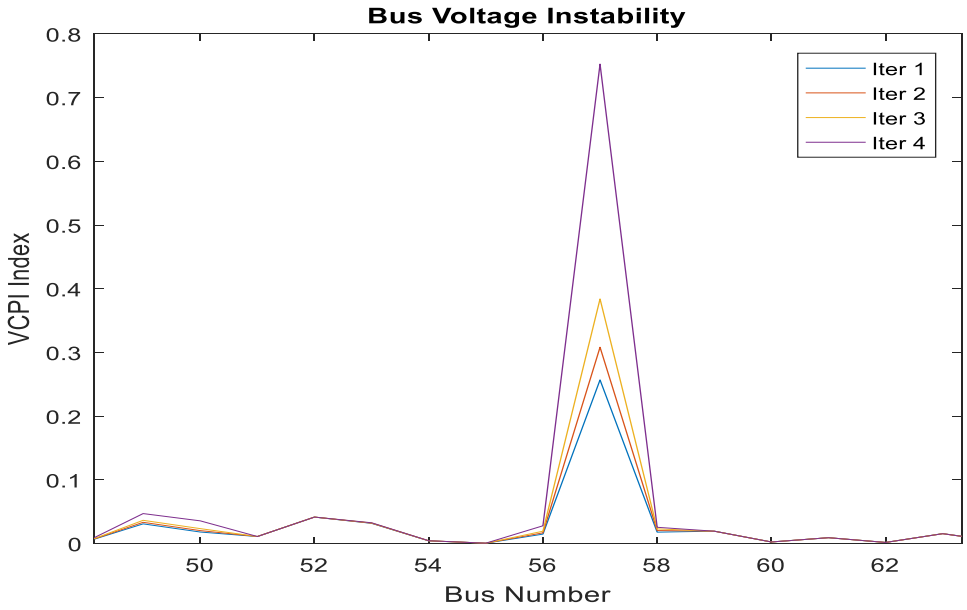

Figure 3-13: VCPI index during equilibrium steps

Figure 3-13 shows the Voltage Collapse Point Indicators (VCPI) index [72] of the buses as the new injected powers were iteratively lowered down towards zero for substation 57. Note that this figure shows

bus numbers and not substations numbers. The bus whose voltage is collapsing is bus number 57 while substation 57 contains bus 56. The maximum decrement in any injection during this iterative smoothing was 8.39 MW. The voltage collapses quicker as it moves towards instability due to the nose curve relationship. This can be seen in this index as it moves towards one. Figure 3-14 shows the exposed line power flows for both the DC and AC scenario after substation 57 has been lost. Note that the line with high power flow was powering the bus that collapsed, and the extra flow is another line that became exposed after the voltage collapse. This bus held a large load that was lost (which can be seen in Figure 3-11), but its removal helped reduce the probability of hidden failures on all exposed lines near the lost substation. This was not an optimal way to reduce these probabilities, but a reconfiguration, re-dispatching, or risk-based load shedding may very well have similar effects.

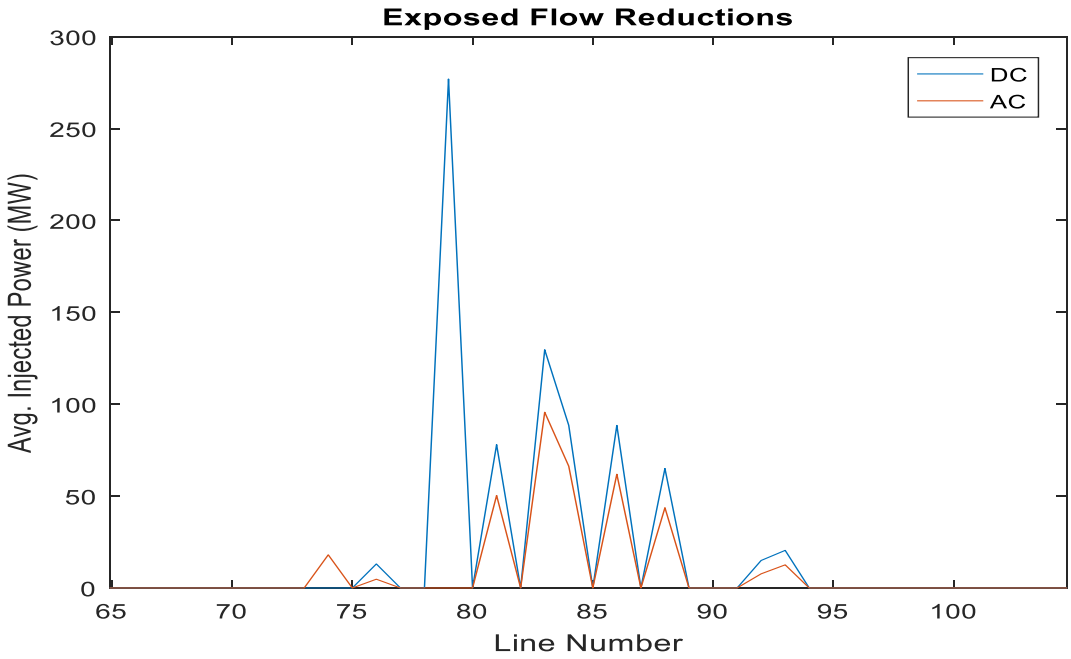

Figure 3-14: Exposed line flow reductions after collapse

Figure 3-15 shows the expected loss associated with a hidden failure at each line while Figure 3-16 shows the top 10 lines' locations on the system. The line from bus 82 to bus 83 has a magnitude of order difference in risk compared to all other lines. It is a prime candidate for maintenance/upgrade of the relay equipment on it. Doing this would most likely extremely lower the overall system risk with one upgrade or maintenance check. Figure 3-17 shows the expected loss-of-load due to cascading for a substation given its removal. These are the same as in Figure 3-11. Figure 3-18 shows the estimated loss-of-load for a system bus due to being involved in all cascades in any substation removal. The dark areas are the least critical and the

yellow areas are the most critical. Figure 3-18 shows that the top left and bottom right have minimum expected loss. Therefore, an entity can prioritize their attentions on upgrading line protections and elements in the high risk areas as per Figure 3-18 while also focusing on substations with high risk as per Figure 3-17.

Figure 3-19 and Figure 3-20 show analogous results using cascading probabilities. They show the substations and buses that have a higher probability of causing a cascade and being involved in a cascade, respectively. Note that the scales are self-normalized to bring out contrast in each graph. Values may not necessarily be comparable, but an overall comparison may still be done. A comparison of Figure 3-19 and Figure 3-20 with Figure 3-17 and Figure 3-18 shows those with the highest probability may not be the ones with the highest expected loss-of-load. However, lines with the highest probability of being involved in cascades do tend to correlate well with their expected loss-of-load and can thus be used as a decent indicator of which line relays to give higher importance in the upgrade and maintenance schedule to maximize reliability. In fact, very few line relays in this system need to be quality checked and upgraded to reduce the expected loss of this system.

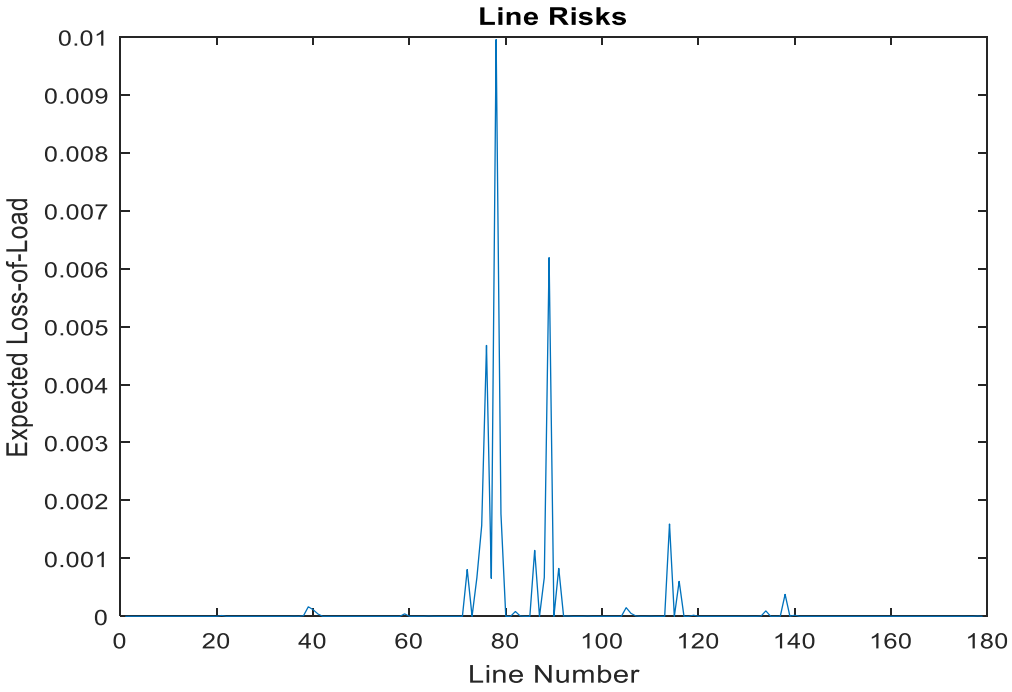

Figure 3-15: Expected load lost associated with each line

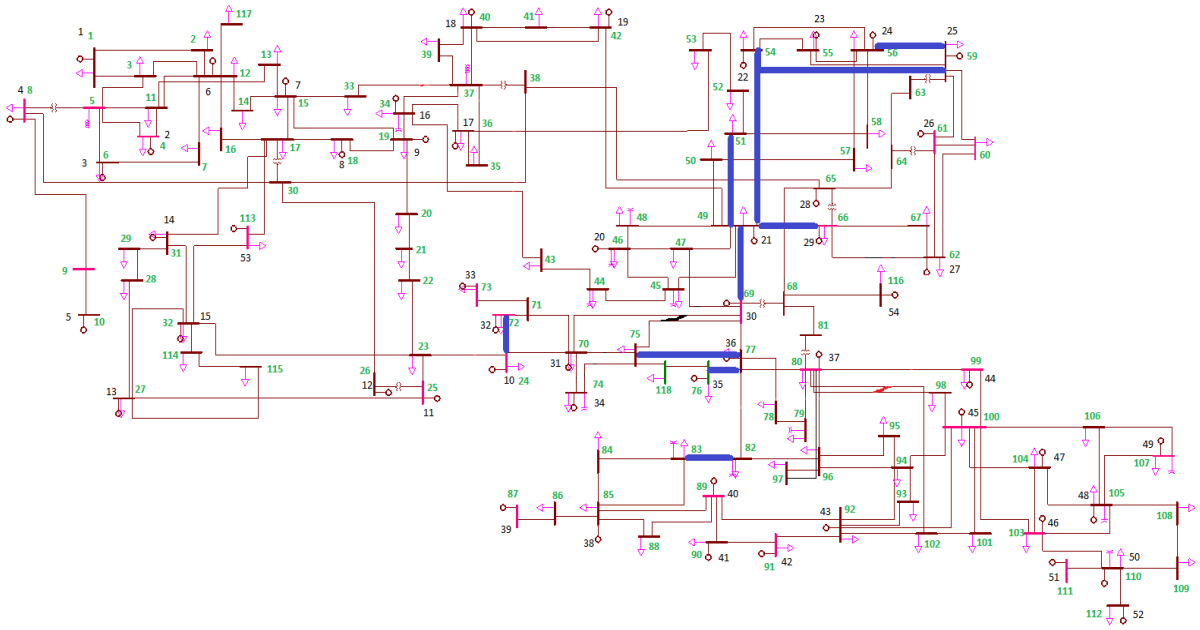

Figure 3-16: Critical Line Locations

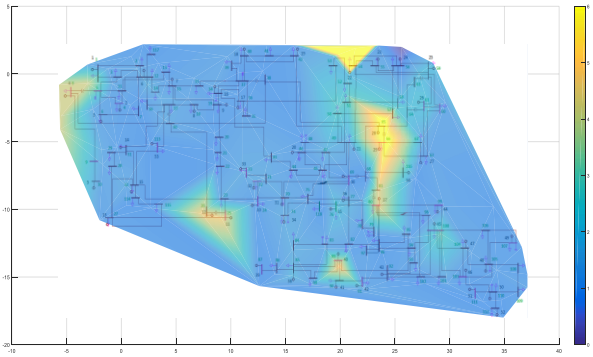

Figure 3-17: Expected loss of load for each substation loss

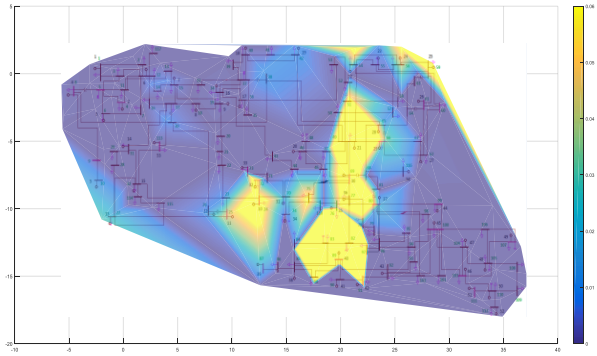

Figure 3-18: Expected loss due to cascade

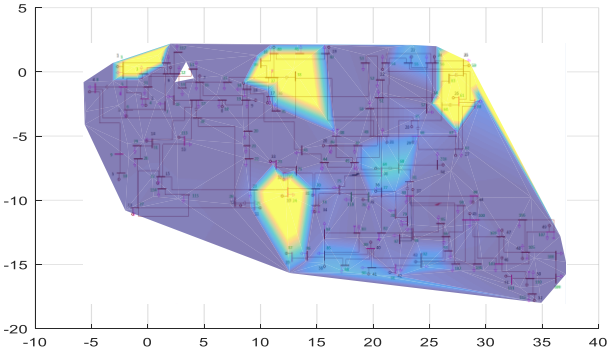

Figure 3-19: Probability of cascading for each substation loss

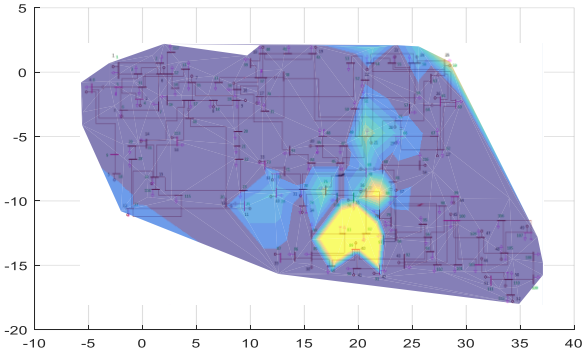

Figure 3-20: Probability of being involved in a cascade

Sensitivity Analysis 1: Misoperation Model

Sensitivity of cascades to the misoperation model was also tested. As this model includes misoperations beyond zone three of relays, it may be conservative and find a much higher risk to substations than a model that does not have misoperations beyond a zone three's threshold. To test how robust these results are to changes in the misoperation model, two other types of models lessening misoperations beyond zone three were tried as shown in Figure 3-21. The first of the two increases the exponential decay of probability by a factor of three; the second model completely reduces probability of misoperation beyond zone three to zero. The results of expected loss-of-load due to cascading from these models in shown in Figure 3-22. The results show that the difference in risk is almost negligible. The three models have almost exactly the same risk of cascading associated with each substation with differences in the risk fitting expectation. Smaller probabilities beyond zone three create smaller estimates of the risk of each substation.

There is a general lack of sensitivity of substations' risk to the probability of misoperation outside of zone three. Most relays will only see impedances so high, even during a loss of substation cascade, that they have near zero probability of misoperating in all these models. This may indicate that the simplest model will give similar results and may be a better model to use going forward as it limits the number of cascades to check.

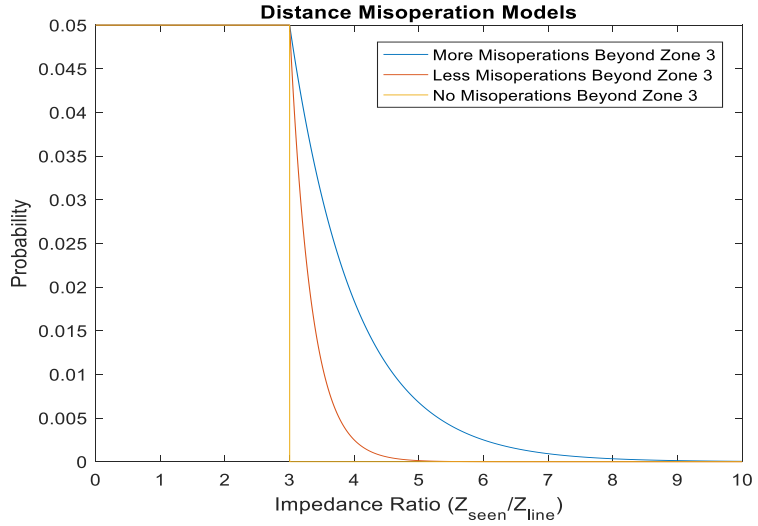

Figure 3-21: Distance misoperation PDF for substation risk sensitivity analysis

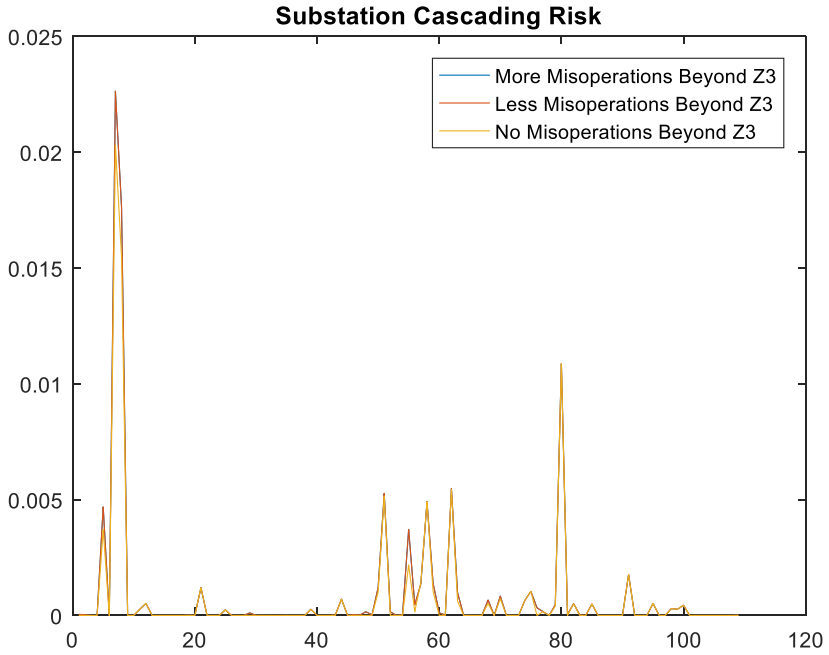

Figure 3-22: Substation risk with varying misoperation models

Sensitivity Analysis 2: System Operating Point

The sensitivity of substation risk is dependent on the loads it is serving or the flow of power through the system. Because of this, analysis was done to see how substation risks changed when the loading of the system changed. Since this model can be computationally expensive, only two other operating points were chosen. It is well known that the IEEE-118 bus system can be split into three distinct areas. In this work, the angular coherency was used to group the buses into their three separate areas. These areas are shown below in Table 3-2 by bus numbers.

Table 3-2: Area Bus Numbers

Area One	**Area Two**	**Area Three**
1,-38, 43,44, 113-115, 117	39-42, 45-81, 97, 116, 118	82-96, 98-112

The first operating point decreases the loading in area one and three to 67% of the original loading while increasing loading in area two to twice its original value. The second operating point decreases loading in area one and area two to 67% and increases the loading in area three to twice the original value. Optimal

power flow was then run on both these loading profiles to obtain new generator dispatches and cascading analysis run. The results can be seen in Figure 3-23 and Figure 3-24.

In general, expected loss-of-load is sensitive to the loading of the system. If an area has higher loading, the substation risks in the area will increase as both probability of misoperation and severity near a substation loss will increase. Depending on how severe the loading change comparison is between substations and areas, it may be enough to change the order of which substations are the most critical. Substation 91 was originally slightly above the "noise level" of the rest of the substations and became most critical substation by risk when its loading level increased. However, most of this substations risk came from load served. Substations two and four had very little expected loss-of-load with almost none from cascading originally, but as loading in the area increased, their risk from cascading increases to approximately 0.0075 and 0.05, as shown in Figure 3-24.

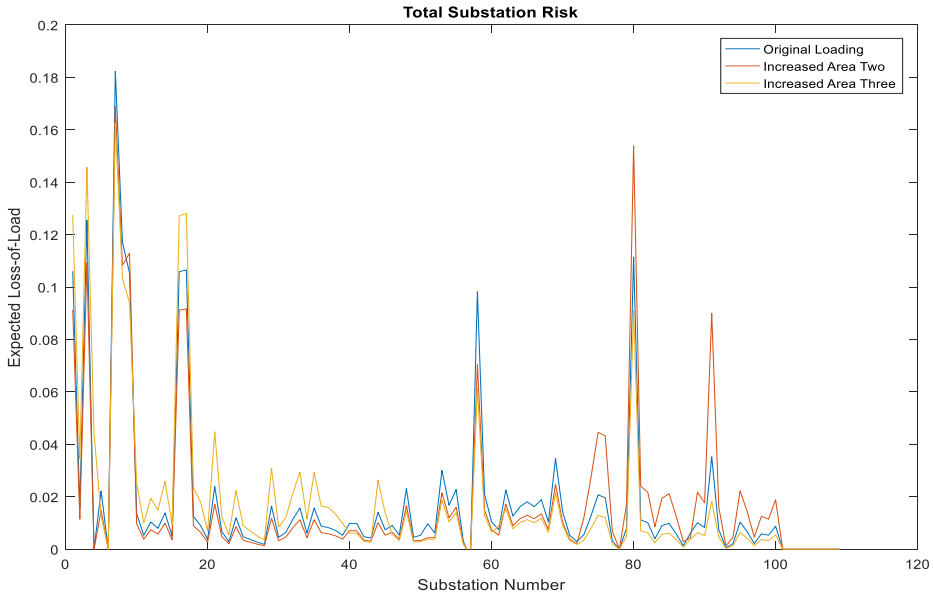

Figure 3-23: Substation Risk under different loading conditions

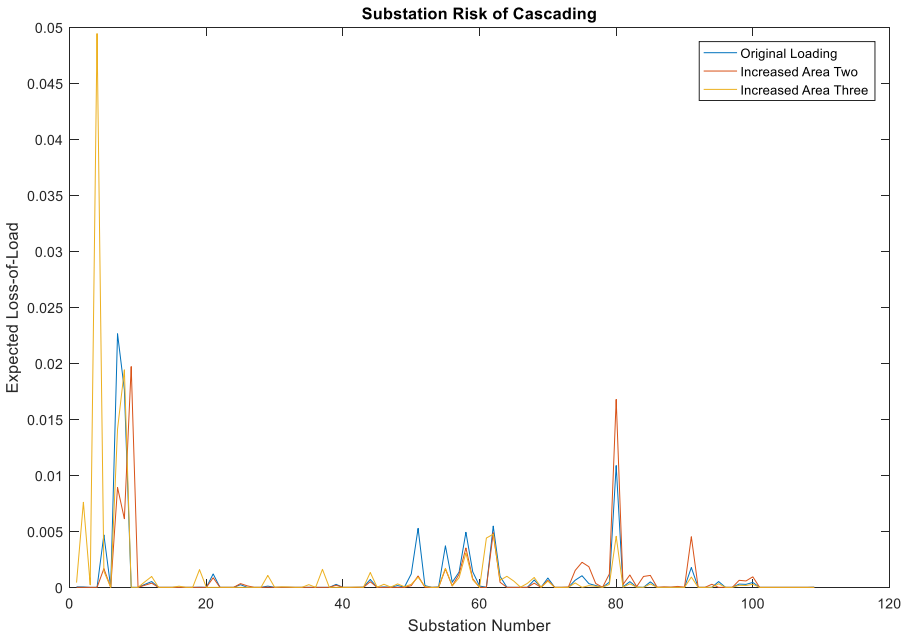

Figure 3-24: Substation risk due to cascading under different loading conditions

This indicates that it may be useful for a utility to perform a seasonal cascading analysis depending on how much the loading profile of the system changes. If the loading profile rarely changes, or changes in a uniform fashion, this analysis may only need to be done for one loading profile to find the most critical substations. However, if loading changes and power corridors change throughout the year, it may be wise to do multiple cascading analysis with representative loading profiles. Since the overall ordering of the risk is robust to loading changes, a representative seasonal loading profile may work for risk assessment.

4 TRANSIENT STABILITY PREDICTION

The previous chapter ignored transient stability of generators during cascades. As this is a large N-k event, transient stability is of bigger concern and one where transient instability may play a larger role. In this chapter, these will be included through the use of a supervised learning algorithm to classify whether a machine has gone unstable during a cascading event.

4.1 Case and Label Creation

Labels for instability and features need to be created to train RFs as they are a supervised learning technique. In order to get the inputs (features) and outputs (labels) for the algorithm to learn the relationship between them, cascading scenarios had to be created. The cases in this methodology were created using cascades from the previous chapter. The general setup for the machine learning algorithm training in shown in Figure 4-1 below.

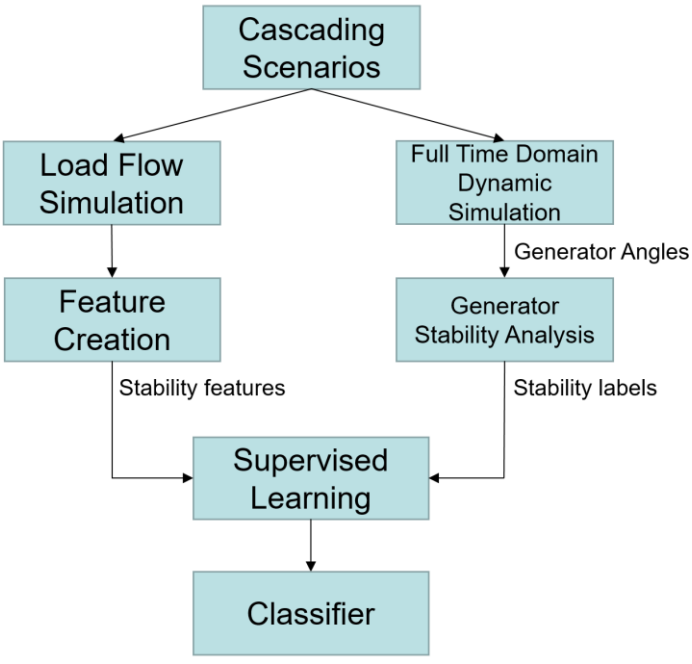

Figure 4-1: Data creation

The cascading scenarios were run in both load-flow as well as time-domain simulations. The time-domain simulations were used to check whether any generators would become unstable. Stability was defined as any generator angle moving more than 180 degrees from the center of inertia angle. The stability responses of each generator were recorded to be used as the labels for the classifier. Tripping events during a cascade

in the time-domain simulation were done after a “quasi” steady-state had been reached from the previous tripping event. The wait period between trips were on the order of 10 seconds. This was done as this is the major assumption under load-flow. The load-flow simulations were used to create the features so that the machine learning algorithm may use those same features in future cascading scenarios to predict whether that machine would go unstable.

4.2 Feature Creation

Feature creation and selection is an important concept in any classification problem. If features that are not indicative of the response or label are used, the classification will have poor accuracy. The algorithm may also suffer from dimensionality when scaling the problem if too many input features such as all system voltages, angles, etc. are given. Simple load flow variables as features will create a classifier with very poor performance, as a lot of information is lost from their dynamic equivalents. Instead, features in this work have been created based on how well they may represent transient instability. These features can be split into three time-based categories: steady-state variables before the event, estimated transient variables during the event, steady state variables after the event, and difference between these values.

The thought process is that variables before and after the event give an idea of how stressed the generator is currently and is going to be afterwards. Incorporating these alone will give a better idea of how close to transient stability the generator may be and increase the classifier performance some. To further increase the classifier performance, a simple, reduced, 2^{nd} order electromechanical model is run briefly after every event to understand what the general dynamics of the system will be immediately after the event. The estimated transient variables give an idea of how the system acts during the event. These sets of variables are listed in Table 4-1. Note that change in predictors is simply a difference between variables before and after the event to give an idea of how severely the event affected a generator. The classifer may learn this dependence but giving it access directly to these variables is thought to reduce complexity.

Table 4-1: Input Features

Feature Name	Feature Variable	#
Maximum MVA Capacity	P_{max}	1
Terminal Real Power	P_g	2
Electric Real Power	P_e	49
Electric Reactive Power	Q_e	50
EMF Voltage magnitude	E_m	3
EMF Voltage angle	E_a	4
Synchronizing Torque	$K_s = \frac{V_s V_t}{X} cos\delta$	5
Driving Impedance	Z_{ii}	6
Critical clearing time (SMIB)	t_{crit}	7
Electric Real Power	$P_{e_{trans}}$	51
Electric Reactive Power	$Q_{e_{trans}}$	52
Terminal voltage magnitude	$V_{m_{trans}}$	8
Terminal voltage angle	$V_{a_{trans}}$	9
Synchronizing Torque	$K_{S_{trans}}$	10
Driving Impedance	$Z_{ii_{trans}}$	11
CoI angles	$CoI = \delta_i - \delta^{COI}$	12
Transfer Impedance	Z_{ij}	13
Generator Speeds	ω_g	14-18
Generator Angles	δ_g	19-23
Area CoI angles	CoI_{g_i}	24-38
Speed derivatives	$d\omega/dt$	39
CoI referenced speeds	$\omega_{COA} = \frac{CoI^{(t+1)} - CoI^{(t)}}{\Delta t}$	40-42
Change in predictors	$\Delta variables$	43-48

4.2.1 Steady State Features

All generator local voltage and power variables were extracted to be used for prediction purposes. The driving impedance and transfer impedances were obtained from the bus impedance matrix. The driving impedance is a measure of how well-connected the generator is to system, and the transfer impedance is a measure of how distance the current event is from the current generator in electromagnetic terms. As a cascade

many buses involved, transfer impedances were defined as the smallest element in the bus impedance matrix from the generator bus to the current event's buses.

Critical clearing time is based on the equal-area criterion of synchronous generators [63] and indicates how much time a fault is allowed to exist at the terminals of a generator before the generator goes angularly unstable. The critical clearing time estimate in this work assumes that the current terminals of the generator were connected to an infinite bus for ease of calculation and computation during the cascading model. Further predictive power may be obtained if a two-machine equivalent is found instead. Synchronizing torque is a measure derived from the linearized equations of dynamic power system models that indicates how much torque there is connecting a generator to the system [73].

4.2.2 Transient Features

All models were assumed to have no governor or exciter response when calculating transient values with the generator itself being a classical model of a constant source behind a transient reactance as shown in Figure 4-2. Typically when estimating transient stability, generators are modeled with a classical model, loads are converted to constant admittance to make the system linear, and the system admittance matrix is reduced to a generator model through a form of Kron Reduction [69]. These assumptions are under the basis that the study time-scale is on the order of a few seconds where the energy in the inertia of the large rotating masses dominate any other control, and many controllers have not had time to adjust [69].

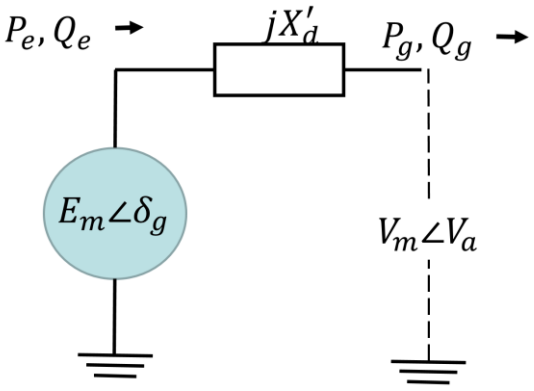

Figure 4-2: Classical Model

A structure-preserving constant PQ load model is used instead of traditional constant admittance models [69], [74] to estimate generator angles and speeds at five consecutive, one-cycle, equal interval

snapshots after the event. In general, the voltages of buses can be described by the system admittance matrix, Y_{sys}, the bus voltages, $V_m \angle V_a$, and the currents injected at each bus, $I_{inj.}$

$$Y_{sys}V = I_{inj} \tag{4-1}$$

We may combine the internal generator buses by appending new admittance terms that include the internal generator voltages in equations (4-2)-(4-3).

$$Y_{sys}V + Y_{tf}E = I_{load} \tag{4-2}$$
$$Y_{tf}^{T}V + Y_{g}E = I_{gen} \tag{4-3}$$

Equation (4-6) gives these in matrix form.

$$\begin{bmatrix} Y_{sys} & Y_{tf} \\ Y'_{tf} & Y_g \end{bmatrix} \begin{bmatrix} V_m \angle V_a \\ E_m \angle \delta_g \end{bmatrix} = \begin{bmatrix} I_{load} \\ I_{gen} \end{bmatrix} \tag{4-4}$$

Generator injections at the terminal buses are turned to zero, and those buses become PQ buses. Here Y_g and Y_{tf} are the self and mutual admittance matrices, respectively, for the internal generator buses. However, since constant PQ load models are used, I_{load} is dependent on the voltage at those load buses. Hence a traditional load-flow Guass-Seidel using all voltages, $V_{adj} = [V\ E]^T$, and power injections, $S_{adj} = \left[P_{load}\ Q_{load}\ P_{gen}\ Q_{gen}\right]^T$ to solve for new bus voltages, V, in replacement of equation (4-2) is used. Once the new bus voltages are calculated, these can be used to calculate I_{gen} in equation (4-3) and the power, real and reactive, injected by the machine. The electromechanical differential equations given by the swing equations in (4-5) and (4-6) as well as the time assumption of one cycle per iteration are then used to obtain new generators' rotor speed and angle.

$$\frac{2H}{\omega_s}\frac{d^2\delta_g}{dt^2} = P_m - P_e \tag{4-5}$$
$$\frac{d\delta_g}{d_t} = \omega - \omega_s \tag{4-6}$$

These new rotor angles are updated in V_{adj}, and the process is repeated for each time snapshot. In this way, we can estimate a generators power output, real and reactive, the rotor angle, and any other load flow-based variable in the transient domain. These snapshot angles and speeds are also compared to their generator's respective coherency area's center of angles as well as the overall center of angle at each time instance.

4.2.3 Generator Coherency

Coherency-based features were deemed necessary as each classifier is used for prediction of one generator and decoupled from others. Generator instability, however, is not a decoupled phenomenon. There are cases where the prediction fails with only local generator predictor variables due to this decoupling. As a brief example, a particular generator drifting from the system will pull close, coupled generators toward instability. If the force is strong enough, the drifting generator will pull the coupled generators out of stability with it. In this case, only the classifier associated with the first generator predicts instability in the load-flow framework, as its local variables were indicating instability. Without that generator pulling the others out of stability, they would not have gone unstable. Coherency measure variables seem to remedy this situation somewhat by allowing the decoupled classifiers to tell how close each generator is to its own coherent area and how far it is from other areas as well as the overall system center of inertia. This finding agrees with previous findings that COI-based variables are known as good indicators of system transient stability [33].

4.2.4 Center of Inertia

Center of Inertia (COI) of the power system is a measure of where the system average kinetic and potential is. It is an average of generator angles weighted by their inertia to give a measure for the system's angle state. Instead of referencing machine angles to a single machine, such as the slack bus, a reference transformation using the COI is used. The transformation given below gives a metric for how far from the system's center each generator is and thus a measure of how close to steady state out-of-step instability each generator is.

$$\delta^{COI} = \frac{\Sigma_i M_i \delta_i}{\Sigma_i M_i} \tag{4-7}$$

$$\bar{\delta_i} = \delta_i - \delta_{COI} \tag{4-8}$$

Similar transformations are done for each coherency group of generators to give a final measure of how close a generator is to the overall COI, its coherency group COI, and other coherency group COIs.

4.2.5 Proper Orthogonal Decomposition

Generator coherency was found using a Principle Component Analysis (PCA) technique called Proper Orthogonal Decomposition (POD) [75]. POD is used to calculate the Proper Orthogonal Modes (POM). The mechanical modes of the system and coherent areas can be approximated from these POMs [75]. The goal of

POD is to approximate some function $z(x,t)$ as a finite sum of coefficients, a_k and orthonormal basis functions, ϕ_k.

$$z(x,t) \approx \sum_{k=1}^{M} q_k(t)\,\phi_k(x) \tag{4-9}$$

Now let us take generator angle measurements at N equally spaced intervals for M generators and combined them in a matrix A such that the matrix A is N x M. We then mean center each column of A and perform a Singular Value Decomposition (SVD).

$$A = U\Sigma V^T \tag{4-10}$$

Here, U and V are both orthogonal matrices of size NxN and MxM, respectively. Σ is a rectangular NxM diagonal matrix. The diagonal elements of Σ, Σ_{ii}, are called the singular values of A, and the number of nonzero diagonal elements is equal to the rank of A. Allowing $Q = U\Sigma$, (4-10) is rewritten as (4-11).

$$A = QV^T = \sum_{k=1}^{M} q_k v_k^T \tag{4-11}$$

Comparing (4-11 with (4-9, the SVD gives the decomposition sought. Note that the singular values are related to eigenvalues of the auto-correlation matrix $X = \frac{1}{N}AA^T$ through, $\sigma_i = \frac{\lambda_i}{n-1}$ and as such give a measure for how much energy is contained within each mode.

Generator dynamics are excited through the use of three phase faults in the 54 machine, IEEE 118-bus system. The linear modes of oscillation are extracted from the columns of Q and further analyzed to gain the system electro-mechanical modes as shown in Figure 4-3. There are two electromechanical modes of oscillation, one at approximately 0.38 Hz and another at 0.53 Hz. The participation of each generator are gotten from the mode shapes in the columns of V as shown in Figure 4-4 and Figure 4-5. In the lower frequency electromechanical mode, there are two areas swinging against one another, and in the higher frequency mode, there are three coherency groups of generators swinging against one another.

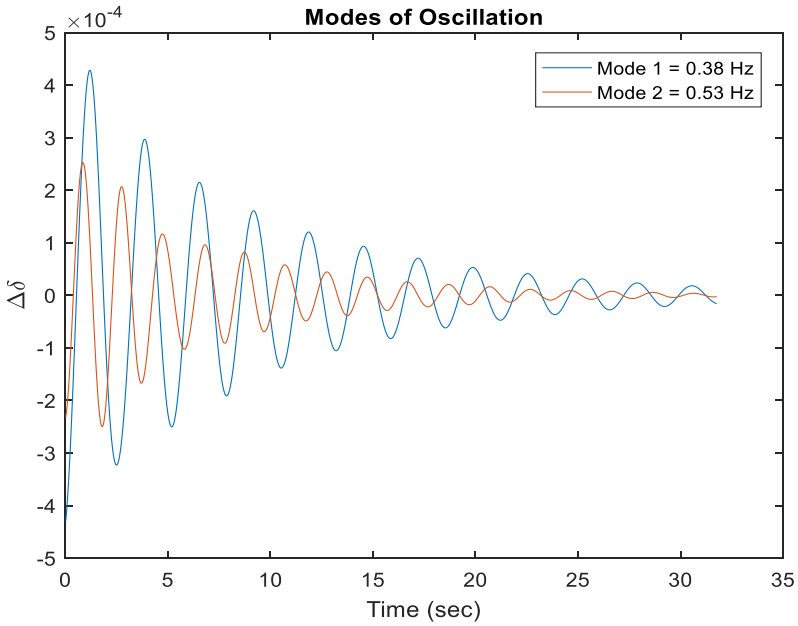

Figure 4-3: Electromechanical Modes of Oscillation

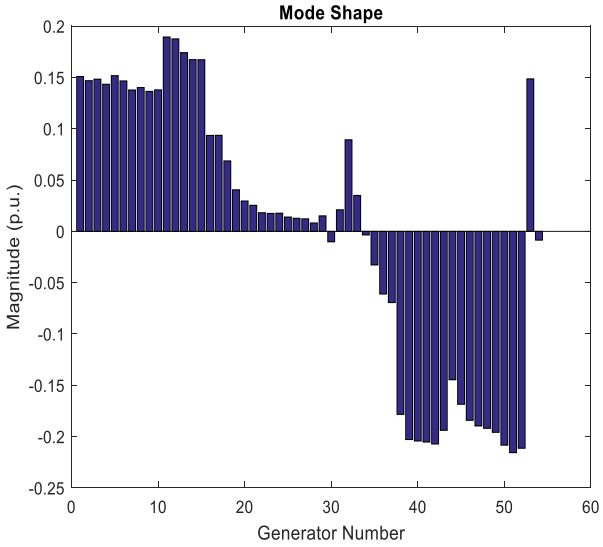

Figure 4-4: Mode 1 Generator Mode Shapes

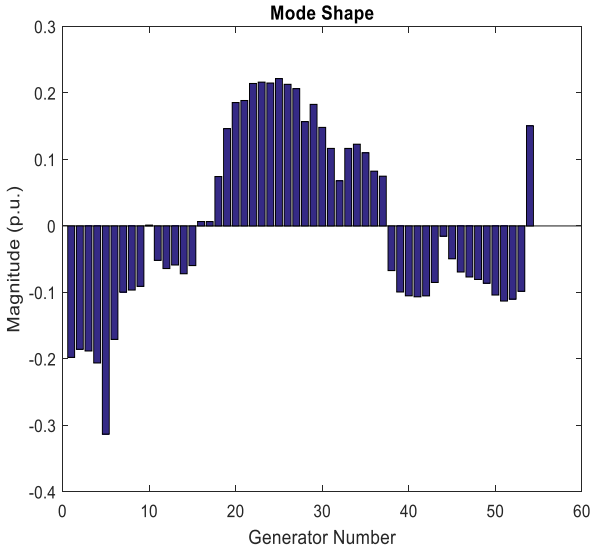

Figure 4-5: Mode 2 Generator Mode Shapes

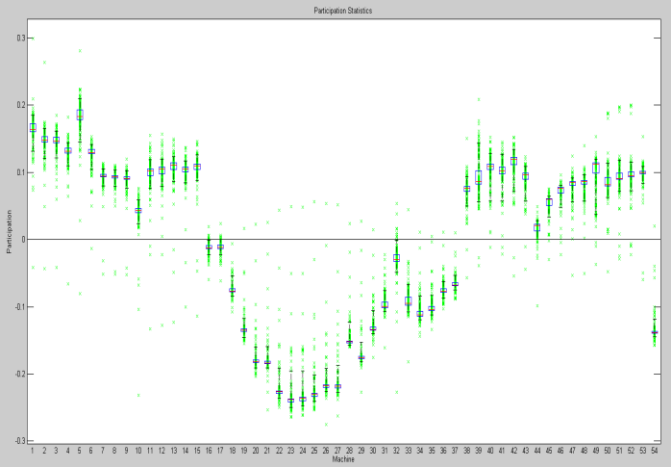

Figure 4-6: N-1 Contingency Mode Shapes (5/95 Percentiles)

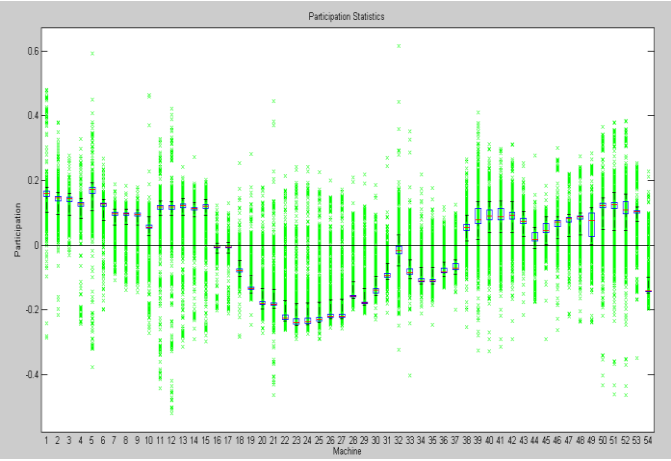

Figure 4-7: N-2 Contingency Mode Shapes (5/95 Percentiles)

N-1 and N-2 analysis was then done on the system to see the effect of contingencies on the coherency groupings of the generators. The results across all contingencies are shown in Figure 4-6 and Figure 4-7 for N-1 and N-2, respectively, with the 5/95 percentiles as well as the median magnitude of participation. The wide variance that accounts for less than 10% of the contingencies in Figure 4-7 is accounted for by nearby contingencies to the current generator. Goherency groupings do not drastically change for generator clusters away from the current cascade. As the cascade gets to a cluster and starts to separate lines between them, etc., the cluster's coherency falls apart.

4.3 Skewed Data

As power systems are designed to be reliable, cases in which generators go unstable are very rare. Out of the cases run in time-domain, few generators went unstable less than 1% of the time with most generators being a magnitude of order less. This skew in unstable and stable cases makes data classification a harder problem for generator instability. A classifier could maintain that nothing goes unstable and still have a reported accuracy of 99%+. There are a few methods in machine learning for getting around this issue: cost-sensitive learning, sampling and synthetization [76]. In cost-sensitive learning, one skews the cost of misclassifying the minority class (unstable) higher than misclassifying the majority class (stable) depending on how much more important the minority class is [77]. Sampling-based techniques teach the classifier the importance of the minority class by over-sampling it or under-sampling the majority class [78], [79]. Synthetization techniques aim to extrapolate the minority cases by creating new synthetic data by extrapolating from existing data [76], [80].

Each technique has its advantages and drawbacks. Under-sampling decreases learning time but potentially discards useful information in the majority class. Over-sampling makes overfitting much more likely in the minority class. A classifier may learn boundaries in the feature space on disjoint sets created by oversampling that do not actually exist resulting in low predictive accuracy. In fact, it is common for a rule to be set for a single oversampled example [80]. Synthetization techniques make an assumption that extrapolation on current data is possible. If the minority class is truly in disjoint sets, the classifier will not learn these sets opting for learning a combination of these sets.

4.3.1 Adaptive Synthetic Sampling

In this work, an adaptive synthetic over-sampling technique called Adasyn is used [76]. New data is adaptively synthesized from the existing minority class depending on how hard those minority cases may be to classify. New samples of features and labels are created for unstable generators to approximately match the number of stable cases with the hope that any convex combination of features between two unstable points in the N-dimensional feature space will all have an unstable result as well. For each unstable generator observation, define the ratio of stable generator observations in the K-nearest neighbors, N_{maj}^{i}, to K as R_i and normalize across all $R_i, i = 1, \dots, N_m$ where N_m is the number of minority examples.

$$R_i = \frac{N_{maj}^{i}}{K} \tag{4-12}$$

$$\bar{R}_i = \frac{R_i}{\sum_{i=1}^{N_m} R_i} \tag{4-13}$$

These $\bar{R}_i$ then give a density of how many of the majority classes (stable generators) are near that particular minority observation (unstable generators). Next, define M_U as the difference between the majority class (stable generators) and the minority class (unstable generators). How many synthetic points to include near each minority observation is then determined by equation (4-14).

$$N_{syn}^{i} = \bar{R}_i * M_U \tag{4-14}$$

For each unstable generator observation, x_i, N_{syn}^{i} points are synthesized based on the following steps:

1. Randomly choose an unstable generator observation, x_r^i, in $x_i's$ K nearest neighbors.
2. Generate a random point between these two points based on linear interpolation in equation (4-15).

$$x_i^{syn} = x_i + \alpha(x_i^r - x_i) \tag{4-15}$$

$$\alpha \sim U(0,1) \tag{4-16}$$

3. Do this N_{syn}^i times.

In this way, more points are synthesized around the boundaries between the majority and minority class, and fewer points are synthesized where there is no confusion as to the class (e.g. the middle of a minority cluster). This gets over the weakness of learning rules to account for isolated minority points but assumes that one can linearly interpolate between nearby minority observations. This allows the classifier to learn boundaries in the training data between unstable and stable cases in the feature space versus learning the few results of unstable cases. The resulting classifiers are then tested on the validation data set without synthesized data to determine how well the forests are actually classifying the data.

4.4 Machine Learning Classifier

4.4.1 Decision Trees

In this work, a decision tree-based ensemble method called Random Forests is employed for transient classification. Decision trees are powerful, nonmetric machine learning algorithms that are able to capture nonlinear patterns in data classification and regression with a sequence of questions about the range or category of variables [81]. They tend to scale well to large sample sizes, deal with irrelevant predictor variables, handle missing data, as well as ignore outliers [82]. They also have the benefit of being intuitive for the user to see the process that classifier used to come up with a final decision as it is simply a sequence of variable range or category questions (interpretability). Decision trees do tend to have the drawback of not being able to capture linear relationships between features very well [82]. This is due to the decision tree creating splitting boundaries on feature axes so linear relationships have to be learned in a step-wise manner. They also have the major drawback of being volatile. If the training data is perturbed, the tree can change substantially [82].

Decision trees classify patterns in the data through a sequence of if-then questions. The sequence starts at the root node and works its way in a directed fashion to a leaf node where the classification occurs. Each question splits the data with data $\in$ *Set*, *true/false* or *yes/no* questions that it learns from the boundaries of data in its training set. While training a tree, there are two major questions the tree has to answer at each node: how many splits should occur and how should the data be split on that data. There are many algorithms for

training a decision tree, e.g. ID3, C4.5, CART, C5.0, etc. For this work, CART has been utilized to train each tree in the random forest. CART is a binary tree algorithm (each node splits into two descendant nodes) that turns any decision into binary decisions. At every decision node, the tree splits into two child nodes based on the value of one variable [81]. As building an optimal tree is N-P complete [83], a greedy approach is taken to decide on which variables to split on at each node [84]. To do this, CART utilizes the Gini impurity index to decide how well a split would be on that variable. The Gini impurity is a form of entropy that gives a measure of expected error at a node N if the label was picked from the distribution at that node randomly [81].

$$G(N) = 1 - \sum_{j=1}^{C} p_j^2 \tag{4-17}$$

Here, p_j is the conditional probability of picking class j given that node N has been reached. The goal is then to minimize this impurity so that these conditional probabilities moves toward zero. In other words, this shows how balanced the classes are at a node. Figure 4-8 shows the Gini Impurity at a node, N, for a two-class problem as a function of the conditional probabilities of the classes at that node.

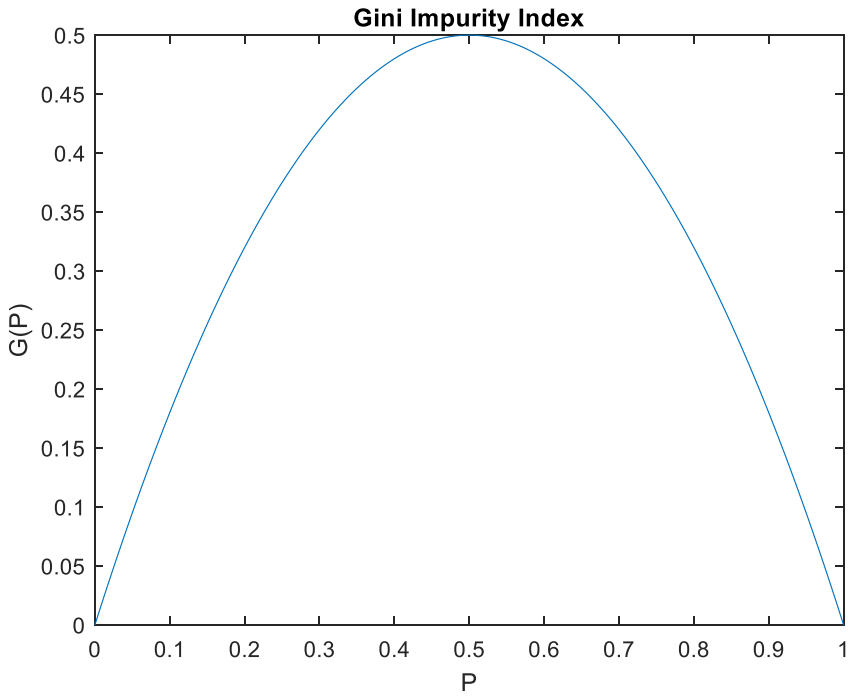

Figure 4-8: Gini Impurity for 2 classes

The minimum value for the impurity is zero if all labels belong to one class, and the maximum value is 0.5 if both classes are equally balanced. Define the current node N with its descendants being N_{right} and

N_{left}, and define p_{left}, the fraction of data that goes to the left node, N_{left}. The drop in impurity from the current node to the two descendants is then given as:

$$G_{drop} = \Delta G = G(N) - p_{left}G(N_{left}) - (1 - p_{left})G(N_{right}) \quad (4\text{-}18)$$

Thus, splits that minimize impurity are chosen in a greedy fashion by maximizing information gain at each node until some stopping criteria such as cross-validation error minimization or impurity decrease thresholding has been reached [81]. The intuition here is that making a decision that makes the classes in each of its descendant nodes as pure as possible will require less decisions to further split the data leading to a small tree.

Impurity decrease thresholding and cross-validation error minimization are forms of generalization to stop the decision trees from overfitting to the training data and performing poorly on any other data. Another method of generalization is pruning. Pruning avoids stopping conditions that end too early due to bad stopping criteria [81]. A tree is first fully-grown to the point where all leaves in the tree have minimum impurity. Pruning then looks at sibling leaf nodes with a common ancestor node to merge. If the merging of two leaf nodes results in a small enough gain in impurity, the two are eliminated and the ancestor node becomes the representative leaf node [81].

4.4.2 Random Forests

Random Forests (RFs) are a specific implementation of bootstrap aggregated (bagging) ensembles using decision trees as base learners [85]. Bagging is one of the earliest, simple ensemble-based algorithms and can be very effective [86]. Bagging is a method that employs N different bootstrap samples (sampled with replacement) to train N different independent classifiers or base learners. Variance in training data between each base learner, as well as making each learner weak, is used to attempt to create diversity between the classifiers to create independence. The decision boundaries between each of the learners should vary notably when the training data is perturbed.

Good examples of weak learners are linear classifiers that have trouble overfitting training data, such as decision stumps (decision trees with one decision), linear SVMs, and others [86]. In general, bagging also allows for stronger classifiers to learn more discrepancies in data boundaries between each learner but loses correlation. Note that a classifier that attains very low bias on a small subset of the training data (strong learner with respect to the subsample) is not a strong learner in general. In general, there will be a trade-off between

the strength of each individual learner and the correlation between learners. Picking larger amounts of data will increase the strength of each individual tree but also increase the correlation between them. The number of training samples to pick becomes a hyper-parameter for the ensemble, but a general good starting point is $\sqrt{N_{train}}$ where N_{train} is the number of observations in the training data.

These N learners are then aggregated in the prediction step through averaging for regression problems and majority voting for classification problems [85]. Thus, each weak learner learns a boundary on their subset of data and these boundaries are averaged together to create the final classifier. Bagging has the advantages over a single classifier of increasing the classifier stability and accuracy when predicting and reducing the classifier variance [82]. By aggregating the prediction results of classifiers trained on random subsamples of the training data, noise and variance is reduced through averaging. If the errors between the N learners are uncorrelated, the average error of the ensemble is provably reduced by a factor of N. However, the errors between the learners are rarely uncorrelated so a complete reduction of error by a factor of N is rare [87].

Figure 4-9 shows an example using MATLAB's fisheriris data considering three classes: setosa, virginica, and versicolor. In total, there are 150 examples. Fifty decision trees are trained on a random subset (sampled with replacement) of 15 observations of the original data. The first 15 tree decision boundaries as well as the data used for that tree are shown in the first 15 graphs. The final graph shows the quantized average border of the results on all of the original data. By creating 50 decision tree stumps and averaging their result, the resultant classifier learns a much better decision boundary. Note that one decision tree may learn this same boundary or better, but will not be as robust to variance and error in the data as the bagged ensemble. It will also be much more prone to overfitting if not trained and pruned correctly.

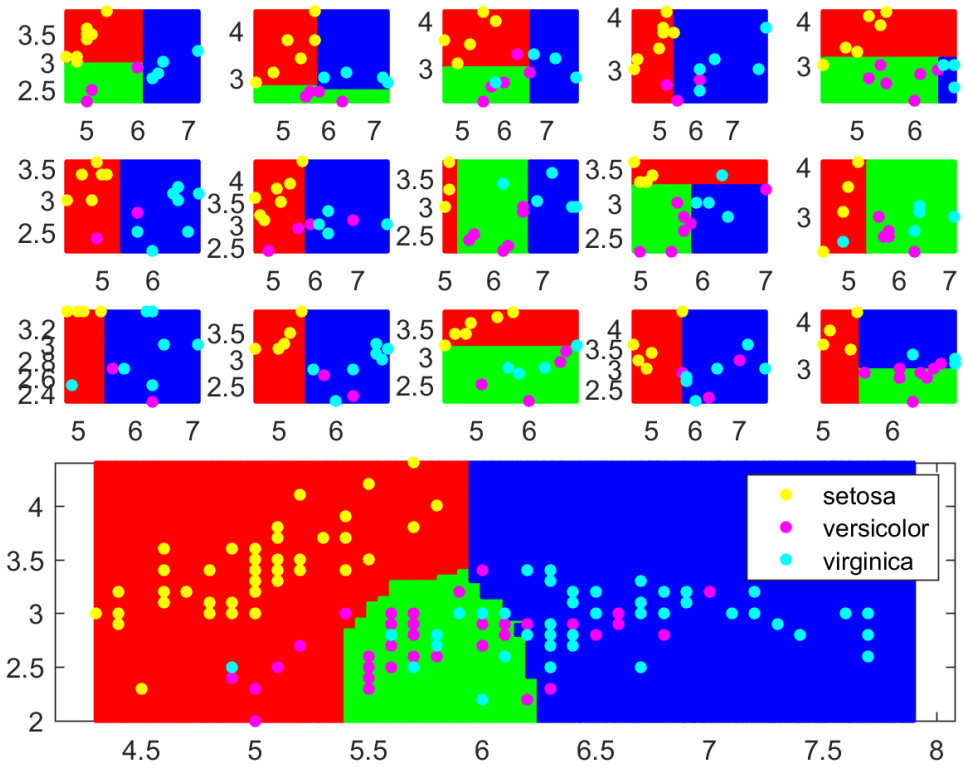

Figure 4-9: Bagging Example

Random forests tend to train large deep trees so that the bias of each individual base tree in the forest is very low on its subsample of training data (unpruned trees) so that intricate details of the subset boundaries may be learned. Forests go a step further in attempting to create independence between trees by adding a feature sampling at each decision node of each decision tree [82]. Hence, each tree only has a subsample of all training data and each node in each decision tree only has a subsample of all features of the data. These layers of randomness add noise to the training data for each tree and tend to make random forests more robust against overfitting to the training data [85]. Since each base tree has been over-trained on a subsample of the data with each node only having access to a subsample of all possible features, it is hard for any given tree to learn true intricacies to the training data. The variance of the forest is then reduced through the aggregating of the decision tree predictions through a voting scheme given in Equation (4-19).

$$f(x_{pred}) = \underset{y}{\text{argmax}} \sum_{k=1}^{N} I(h_k(x_{pred}) = y) \tag{4-19}$$

Here, $I(x)$ is the indicator function that returns 1 if x is true and 0 if x is false. These forests are appealing as they help overcome the volatility of decision trees to input data [33] and provide their own form of generalization error as well as a measure of feature importance [82].

4.4.3 Out-of-Bag Error

Random forests have the quality, in contrast to most machine learning models, as model complexity increases (more trees are added), the less likely the forest is to over-fit [85]. However, the forests can and do sometimes over-fit in certain cases [88]. To avoid over-fitting in machine learning algorithms, some form of generalization is used. This is usually some form of cross-validation or algorithm specific way to generalize the prediction and avoid learning noise within the training data [89]. Random forests make use of out-of-bag (OOB) estimates as a form of generalization error. These can used to determine how large each forest should be to avoid overfitting in RF.

OOB data is the data that did not get selected as part of the bootstrap sample for each tree. Thus, these can be useful for some form of validation to check to see that the forest is not overfitting to the training data. However, the result would not be realistic if all tree predictions were used for each OOB data sample as some trees were trained on that data sample. Only the predictions from the trees that were not trained on that training sample (OOB predictions) are used for validating the generalization error [82]. [85] has found that heuristically this estimate is unbiased in many tests though no proof exists. These OOB errors can be used as the forest grows to determine when the validation error would start to saturate without actually using the validation set. Figure 4-10 shows the results of these runs for all 54 random forests (one for each generator) for the IEEE-118 bus system. Most generator forests need approximately 15 trees before the OOB error starts to saturate. However, there are a few that need almost 50 trees before starting to saturate. Note that these errors were performed on the synthesized, balanced data from the ADASYN algorithm. As such, the overall OOB saturation error magnitude is meaningless with respect to the original data, though it is indicative of when the forest is not learning anymore.

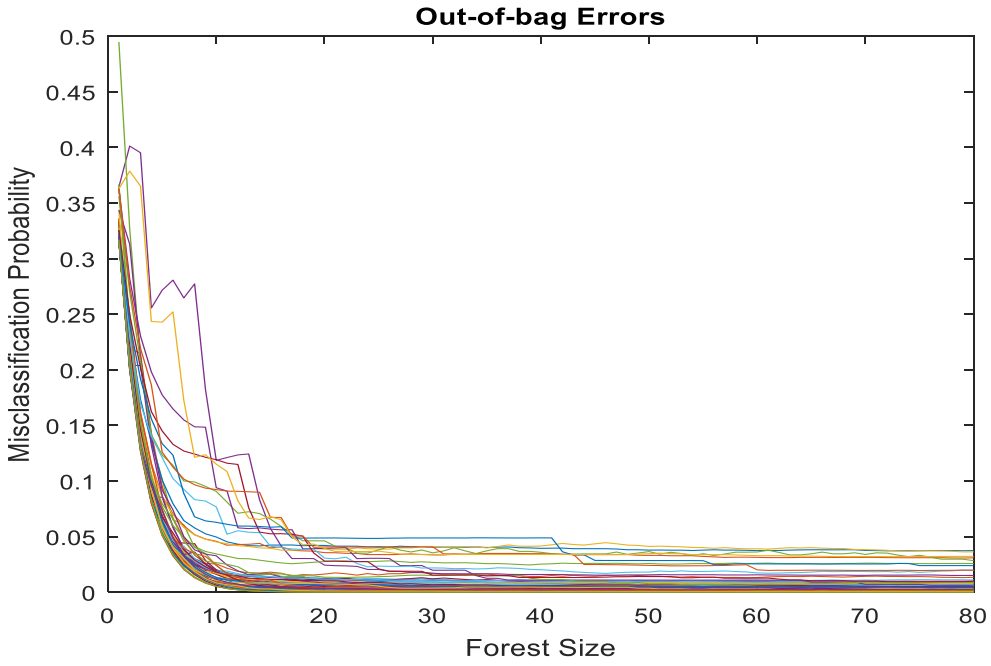

Figure 4-10: Out-of-bag errors as tree size increases

4.5 Feature Importance

As non-indicative features cannot help and might make classification performance worse, feature permutation of the forests using interaction tests [90] and standard Classification and Regression Trees (CART) algorithm [91] was used to determine how important each feature was to the overall performance of each forest. These two methods were compared against one another as a sanity check of variable importance. Interaction tests are statistical χ^2 tests that minimize the ρ-value between each feature and the label as well as between each pair of features and the label [90], [92]. In other words, they test the hypothesis that there is no interaction between a pair of features and the response label. During each decision tree node split, the algorithm picks the variables with the minimal ρ-value that are most likely for the response to be dependent. CART, on the other hand, picks the variable split that maximizes the Gini, or some related entropy, index for each decision node [91].

Feature permutation takes the values of each feature and randomly permutes the values to check the increase in prediction error of the forest. If the prediction error increases significantly, that feature is considered of high importance, and if the prediction error does not change much, the predictor had little effect on the accuracy of the forest. The intuition here is that if a tree did not use a feature to determine the classification, then the feature's value should not matter. The same OOB errors are used as they are generally

unbiased and to avoid touching the testing set. The results are shown in the boxplots in Figure 4-11. Prediction error differences are obtained for each tree, averaged across all trees in each forest, normalized by their standard deviation, and then averaged across all forests across all generators. These features are ordered as in Table 4-1 for labeling of the variable. As the exhaustive search (CART) method and the statistical (interaction) method tend to agree on variable importance ordering, these importance orderings are assumed to be the actual ordering to be used for feature selection.

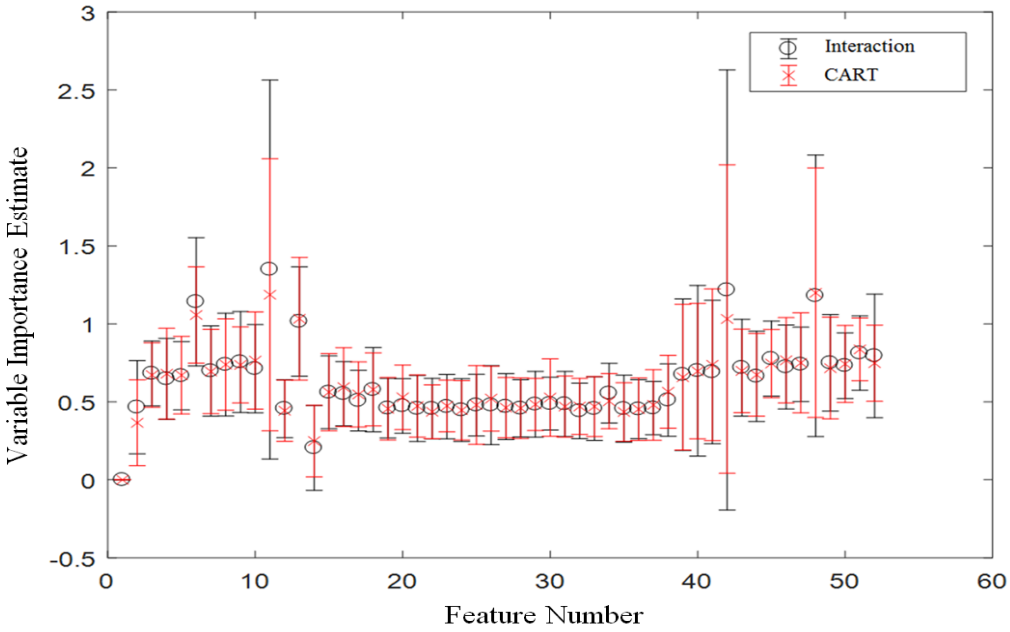

Figure 4-11: Feature importance by CART and Interaction

The two least important variables by mean are variable 1 and 14, generator capacity and estimated generator speed one cycle after the event. This is expected as speed has not had time to change immediately after the event due to the inertia of the machine and capacity never changes as each generator has its own random forest. The most important variables for all the generators by mean are 6, 11, 13, 42, and 48. These are the driving impedance before event, driving impedance after event, transfer impedance after event, aggregated area speed, and difference between driving impedance before and after the event.

Aggregate area speed for one coherency group also has the highest deviation across different machines due to the coherency issue discussed earlier. Of the three areas in the IEEE-118 bus system, only one area tends to have this problem. Generator instability far out of this area has very little correlation with this feature. Driving and transfer impedances at the machine tend to be important to be able to determine accurately whether it will go unstable or not. It is believed this is due to these events consisting of only line trips. Hence,

no kinetic energy, other than the initial loss-of-substation, is infused in the system and only potential energy reductions occur after any given event. The results of this classifier may not do so well in classifying generator instability due to increases in kinetic energy such as faults but may be retrained with those scenarios in mind.

4.6 Predictive Performance Results

To validate the forests, 50% of the data was held out for testing. The following section is the result of testing the classifiers on that 50%. As the data is highly skewed, metrics such as accuracy and overall error do not give any insight into the performance of the classifier. Instead ROC curves, which combine true positive predictive rate with false positive rate [93], as well as precision and recall graphs are used. ROC curves are estimated by bootstrapping the scores of the classifier (can be interpreted as a probability of being unstable in binary classification) and varying the threshold of which scores should be considered positive to determine the estimated predictive power of the classifier among different operating conditions [94]. A worst case classifier equivalent to random guessing is a diagonal line while a square curve is a perfect classifier. The estimated mean of these ROC curves are shown in Figure 4-12. Many machines have very few positive (unstable) examples resulting in ROC curves with low resolution. The ROC curves show classifiers that seem to be very powerful as most are very close to perfect. However, the data skew forces us to look also at the classifiers' precision and recall.

Precision is the ratio between the number of correctly labeled unstable cases and the number of all cases labeled unstable; recall is the ratio between the number of cases correctly labeled unstable and the number of cases actually unstable. The precision-recall results shown in Figure 4-13 give more insight into the classifiers. The recalls of all machines do very well. Hence, this classifier marks almost all cases where the generator is unstable as unstable. The precision metric, however, suffers from the skewed data. As the ratio of stable to unstable cases is anywhere from 99-999:1, depending on the generator, mislabeling even 0.1-1% of the stable cases as unstable will result in a precision of 0.5. The results can be seen from the confusion matrices shown below with Table 4-2 normalizing over true cases and Table 4-3 normalizing over labeled cases. The classifier has a high overall accuracy of 99.07% and high recall. If a case were actually unstable, the machine would have a 94.35% chance of correctly labeling it as such (recall/true positive rate) while having a 0.93% chance of mislabeling a stable case as unstable (false positive rate). However, if the classifier were to say a case is unstable, there is only a 46.52% chance it is actually unstable (precision).

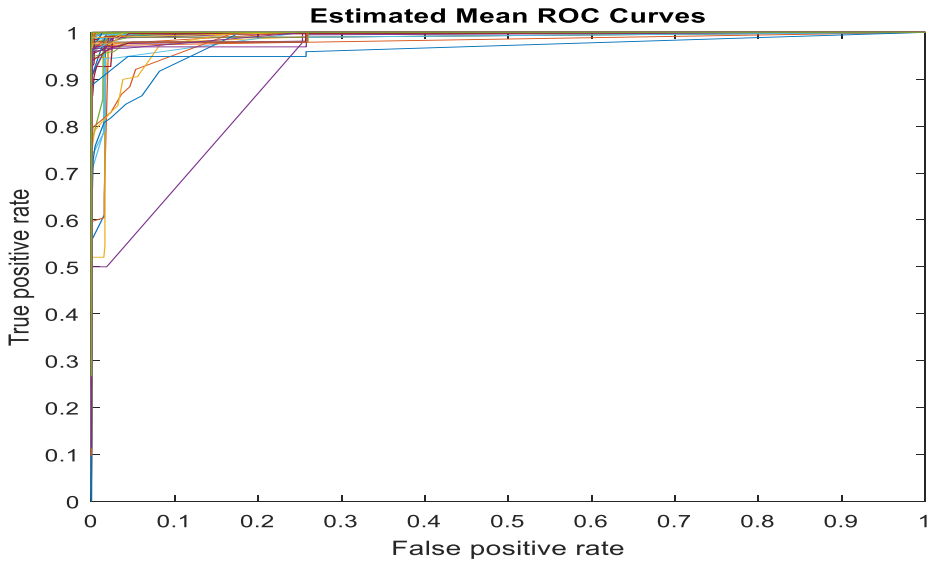

Figure 4-12: Mean ROC curves of all generators

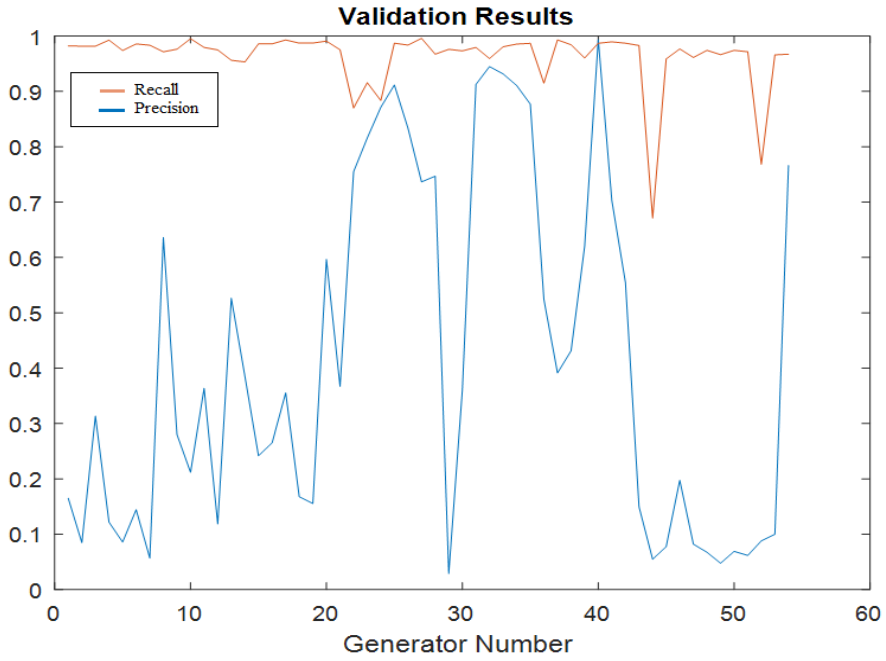

Figure 4-13: Precision and Recall across generators

Different recall-precision results can be obtained by skewing what is important for the classifier to learn. However, there is a tradeoff between precision and recall. The results for a different skew towards stable cases are shown in Table 4-4 and Table 4-5. Average precision across machines increased from 46.52% to 64.73% but at the cost of average recall dropping from 94.35% to 83.58%. As we would like to err on the side of being conservative with the prediction of stability within a cascading type scenario, it is deemed that

recall is more important and the classifier is skewed as such. This gives a classifier that is less likely to predict stable when the system is actually unstable.

Table 4-2: True positive and false negative rates

	Stable Label	**Unstable Label**
True Stable Cases	99.07%	0.93%
True Unstable Cases	5.65%	94.35%

Table 4-3: Positive predictive & false discovery rate

	Stable Label	**Unstable Label**
True Stable Cases	99.95%	53.48%
True Unstable Cases	0.05%	46.52%

Table 4-4: Skewed true positive and false negative rates

	Stable Label	**Unstable Label**
True Stable Cases	99.61%	0.39%
True Unstable Cases	16.42%	83.58%

Table 4-5: Skewed positive predictive & false discovery rate

	Stable Label	**Unstable Label**
True Stable Cases	99.86%	35.27%
True Unstable Cases	0.14%	64.73%

4.7 Cascading Results

Figure 4-14 compares the expected loss-of-load results of the DC, AC, and AC with RF stability prediction models included. Figure 4-15 shows the expected loss-of-load of these models due to cascading. The results are nearly identical with a few minor differences between the transient prediction and simple AC model. Looking at Figure 4-15, the transient stability model tends to overestimate expected loss-of-load for most substations by a miniscule amount. Two very notable exceptions are substation five and seven. This is believed to be a similar phenomenon as the voltage collapse scenario discussed in section 3.5. It is believed that generators tripped that caused relief in lines near the cascade lowering its probability of spreading further. While the decrease of critical rankings due to this impact is negligible (Figure 4-14), the risk due to cascading has been reduced to approximately 80-90% of its original value (Figure 4-15). These two substations' loss-of-load seem to be sensitive and may be reduced through a number of manners as shown in the voltage-based load shedding as well as in Chapter 6.

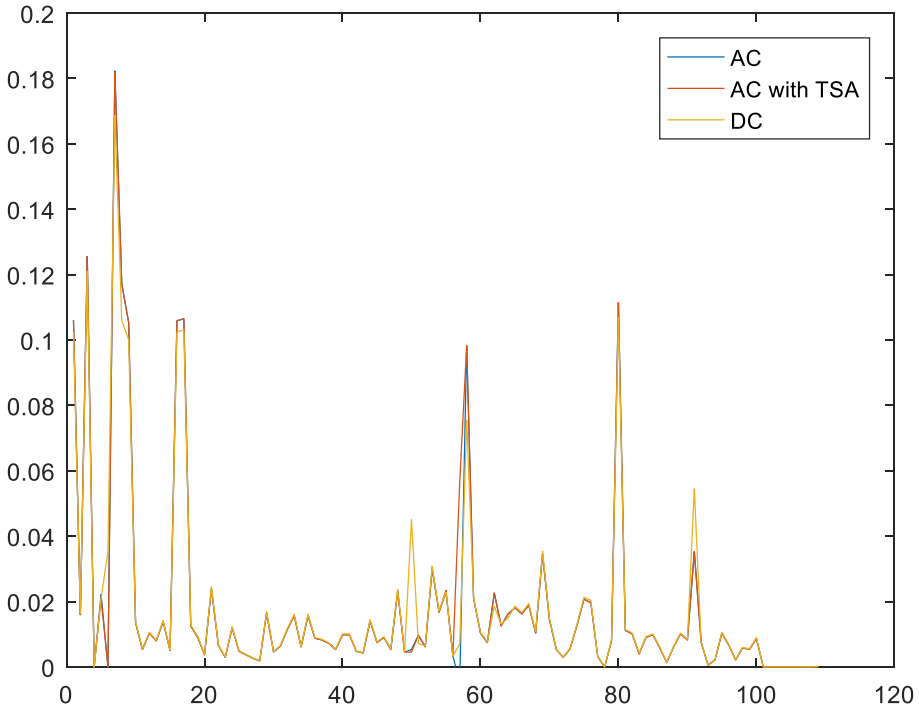

Figure 4-14: Expected load lost under varying models due to substation loss

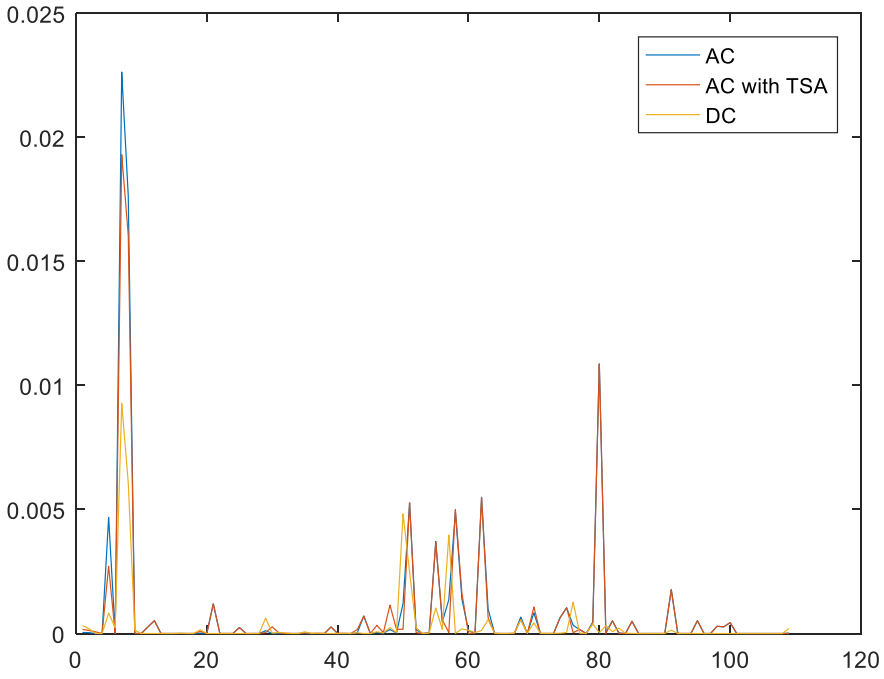

Figure 4-15: Expected load lost under varying models due to cascades

Figure 4-16 examines the differences between the cascading sizes among the varying models. Unique cascades are recorded along with their estimated probabilities and loss of load. Variable binning is then used to estimate the probability distribution function of the blackout size where each bin power loss and probability are computed as the average of power loss and probability within the bin. The difference between models can be seen much more clearly in this manner. The DC model may produce similar substation risk results, but it severely underestimates the probability of larger cascades occurring. The risk threshold in this work was set to 10^{-12}, and the DC model says that at this approximately probability, only 30% of the system load will be lost in the worst cascades.

There is negligible difference between the AC model and the AC model with transient stability. These two have much more shallow slopes compared against the DC model indicating much more critical models. Rather than 10^{-12} for 30% loss, these models estimate a 30% loss to occur with a probability of approximately 10^{-6}.

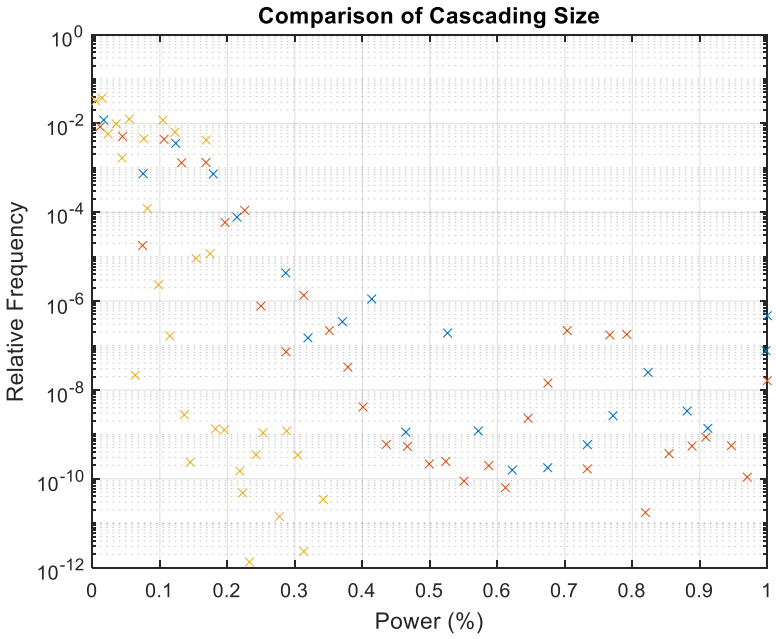

Figure 4-16: Blackout Size Comparison

The average number of misoperations in each model is given in Table 4-6.

Table 4-6: Average misoperation in a cascade

AC Model with TSA	AC Model	DC Model
7.6017	6.8532	6.2696

The average is simply the sum of misoperations in each unique cascade found normalized by the number of unique cascades found. The probability of each cascade was not taken into account for this average as the expectation of number of hidden failures in a cascade will be very close to zero as there is a high probability no hidden failures are exposed. These are higher compared to the major historical disturbances that typically have four to five hidden failure exposures [31]. These numbers would most likely decrease if other initiating failures such as line outages, generator outages, etc. were to be considered rather than simply substation losses. Even though the models tend to give differing probabilities and cascade scenarios, it can be seen that the substation criticality rankings do not change considerably when considering risk due to cascading. If the immediate loss of load due to the initial attack (i.e. load being served by the substation) is also taken into account, the ranking differences becomes almost negligible suggesting that this ranking may be robust to many different factors and lending credence to which substations are the most important.

5 RISK BASED ECONOMIC DISPATCH

Given these cascading analysis results, it is possible to reduce the risk of the system through a preventative dispatch. This dispatch would take increase the cost of operating in order to decrease the risk from cascading due to substation losses. This dispatch could be run in times of doubt over whether substations will be disconnected from the system due to weather or even attacks.

5.1 Optimal Power Flow

In general, an AC OPF tries to minimize the cost of generation, real and/or reactive, while meeting voltage and flow constraints of the system. A standard AC OPF is formulated as:

$$\min_{P_G,Q_G,V} \sum_{i=1}^{N_{gens}} C_{P_{G_i}}(P_{G_i}) \tag{5-1}$$

Subject to

$$V_i\left(\sum_{j=1}^{N} Y_{ij}V_j\right)^* + S_{D_i} - S_{G_i} = 0, i = 1, \dots, N_{buses} \tag{5-2}$$

$$|S_i| - S_{max} \leq 0 \qquad i = 1, \dots, N_{lines} \tag{5-3}$$

$$V_i^{min} \leq |V_i| \leq V_i^{max} \qquad i = 1, \dots, N_{buses} \tag{5-4}$$

$$P_{G_i}^{min} \leq P_{G_i} \leq P_{G_i}^{max} \qquad i = 1, \dots, N_{gens} \tag{5-5}$$

$$Q_{G_i}^{min} \leq Q_{G_i} \leq Q_{G_i}^{max} \qquad i = 1, \dots, N_{gens} \tag{5-6}$$

$C_{P_{G_i}}(P_{G_i})$: Cost of operating generator i at P_{G_i}
P_{G_i} : Generator real power set point
Q_{G_i} : Generator reactive power set point
V_i : Complex voltage at bus i
S_{D_i} : Complex power demand at bus i
S_{G_i} : Complex power generated at bus i

Allow the previous variables to be combined into a vector x. The previous equations may be summarized as follows.

$$\min_x f(x) \tag{5-7}$$

Subject to

$$g(x) = 0 \quad (5\text{-}8)$$
$$h(x) \leq 0 \quad (5\text{-}9)$$
$$x_{min} \leq x \leq x_{max} \quad (5\text{-}10)$$

$$x = \begin{bmatrix} \Theta \\ V \\ P_g \\ Q_g \end{bmatrix}$$

Here, f(x) represents the least-cost generation; g(x) represents the power flow equations and h(x) represents line flow constraints. Finally, x is a $\left(2n_{buses} + 2n_{gens}\right) x\ 1$ or $n_x\ x\ 1$ vector that represents the decision variables. This is a nonlinear optimization problem as the power flow equations and flow limits are both nonlinear. In this work, a trust-region based augmented Lagrangian method, implemented in Matpower, is used to solve this optimization [71].

5.2 Risk Corrected OPF

The OPF may be augmented to help minimize the risk to the system from cascades after a substation loss thereby mitigating substation risk as well. SCED methodologies already exist that minimize base case generation substation subject to base and post-contingency constraints [48]. Utilizing this SCED methodology in cascading situations will be prohibitively expensive due to the combinatorial nature of cascades when including the post-contingency constraints. One way to overcome this is to model only the high risk contingencies that represent a larger threat to the system. If the number of cascade scenarios is large enough, the problem may become infeasible if post-contingency constraints are too many. Another simpler method is to include risk-based indices into the cost function itself of the OPF. In this manner, the OPF will try to minimize the risk as well as the system cost.

Ultimately, the risk of a cascading event in this model is dependent on the apparent impedance each line relay sees. Higher apparent impedances seen by the relays lead to lower probabilities of misoperation. If the apparent impedance seen by each relay during the cascade is increased throughout a cascade, the overall probability of the cascade will decrease. Assuming that the apparent impedance of a line relay during any cascade is correlated with the base-case apparent impedance, system risk may be reduced by increasing the apparent impedance of high-risk line relays in the base case. For simplification purposes, assume that voltage

remains constant and that the real power flow through a transmission line is much larger than the reactive power flow. Then, the apparent impedance of each line relay may be increased by reducing the real power flow through the line.

$$\min_{\theta} \sum_{i=1}^{n_{lines}} Risk_i * P_i \tag{5-11}$$

The aim is to minimize real power flow on lines that have high risk. To do this, a DC power flow approximation is made. Line risks are gotten from the cascading analysis in the form of expected loss-of-load as in equation (3-31). To obtain the power flowing on lines, the power injected at one end of the line is used. As this is a DC approximation, there are no losses so each line injection represents line flow. Consider the construction of a $(n_{lines}\ x\ n_{bus})$ bus-branch incidence matrix, C_{ft}. Each row of C_{ft} represents a branch and each column represents a bus. Element *(i, j)* of C_{ft} is 1 if branch *i* originates from bus *j*, is -1 if it ends at bus *j*, and zero otherwise. Since each branch may only connect to two buses, only two elements of each row of C_{ft} may be nonzero, and the rows must sum to zero. Consider also the construction of a $(n_{lines}\ x\ n_{lines})$ diagonal branch matrix, B, who's diagonal elements *(i, i)* represent the series susceptance of each branch. We may then define a $(n_{lines} x\ n_{buses})$ matrix B_f as below.

$$B_f = B * C_{ft} \tag{5-12}$$

Define a $(n_{lines}\ x\ n_{lines})$ diagonal matrix R whose diagonal entries are the risks associated with each line gotten from prior cascading analysis normalized by the maximum line risk. R allows the creation of risk weighted power injections of lines at from buses as a linear combination of bus angles.

$$P_{f_{weighted}} = RB_f\theta \tag{5-13}$$

The goal then is to minimize the sum of these weighted power injections. However, as some power injections may be negative, the sum of the squares is instead minimized.

$$\min_{\theta} P_{f_{weighted}}^T * P_{f_{weighted}} = \theta^T\left(B_f^T R^T R B_f\right)\theta \tag{5-14}$$

To incorporate this with the traditional OPF, define a new $(n_x\ x\ n_{buses})$ matrix, B'.

$$B'^{T} = R\begin{bmatrix} B_f^T \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5-15}$$

The new minimization function is then given as:

$$\min_x \left[B'^{T}x\right]^{T}\left[B'^{T}x\right] = x^T B' B'^{T} x \tag{5-16}$$

Define the matrix, $H = B'B'^{T}$. Note that H is a block diagonal matrix consisting of a weighted Laplacian matrix of the power system topology graph and a zero matrix. It is a symmetric, positive semi-definite matrix by construction. Adding this to the original cost function will make the new optimization take risk into account. However, as the new operating point moves away from the current operating point, risks associated with each substation and lines hold less. With a new operating point, comes new risks. To stop the optimization from moving too far from the original operating point, a regularization term is added to the cost function. The regularization term is made so that the square residuals of the new operating point's decision variables are not too large, i.e.

$$\min_x (x - x_0)^T (x - x_0) = \min_x x^T x - 2x_0^T x + x_0^T x_0 \propto \min_x x^T I x - 2x_0^T x \tag{5-17}$$

Therefore, we modify the previous matrix, H, with the identity matrix to obtain $Q = H + I$, a positive definite matrix. Finally, define $C = -2x_0$, and the new OPF can be formulated as follows.

$$\min_x (1 - \alpha) f(x) + \alpha(x^T Q x + C^T x) \tag{5-18}$$

$$g(x) = 0 \tag{5-19}$$

$$h(x) \leq 0 \tag{5-20}$$

$$x_{min} \leq x \leq x_{max} \tag{5-21}$$

$$x = \begin{bmatrix} \Theta \\ V \\ P_g \\ Q_g \end{bmatrix}$$

$$0 \leq \alpha \leq 1$$

Each line is weighted by their involvement in the overall risk of the cascade analysis. Hence lines that had higher risk in their cascading scenarios will tend towards lower flows. α is a parameter to tune how much the optimization balances risk versus the least-cost generation. It can be thought of as a parameter to decide how likely the loss of a substation is or how important mitigating those risks are. An α of zero means there is

no chance of losing a substation, and the OPF simply finds the original OPF. An α of one means that a substation will be lost and a new operating point that minimizes the current risks should be found regardless of the cost. The regularization terms term plays the same role as the original objective in order to make the current operating point more stable when α is close to one.

It was found that using the line risks directly was not an effective method. As many line risks are close to zero, the new OPF would reduce risk by increasing the flows on zero risk lines to reduce the high risk lines. Instead, a shifted logarithmic version of the line risks is used including a constant offset of all risks so they remain positive. The function is shifted by its minimum and as a small constant. It is then shifted again to retain positivity and normalized. I.e.

$$R_{new} = \frac{R' + \min(R') + c_2}{\max(R' + min(R') + c_2)} \tag{5-22}$$

Here, $R' = \log(R_{old} + c_1)$ and c_2 is to remain smaller than the minimum of the previously shifted risks. R_{old} are the nonnegative normalized line risks from earlier. It is shifted by c_1 to make sure R_{old} is not zero for the log conversion. This creates line risks that are have the same ordering as previously because the log function is a monotonically increasing function. Figure 5-1 shows the new line risks that were used.

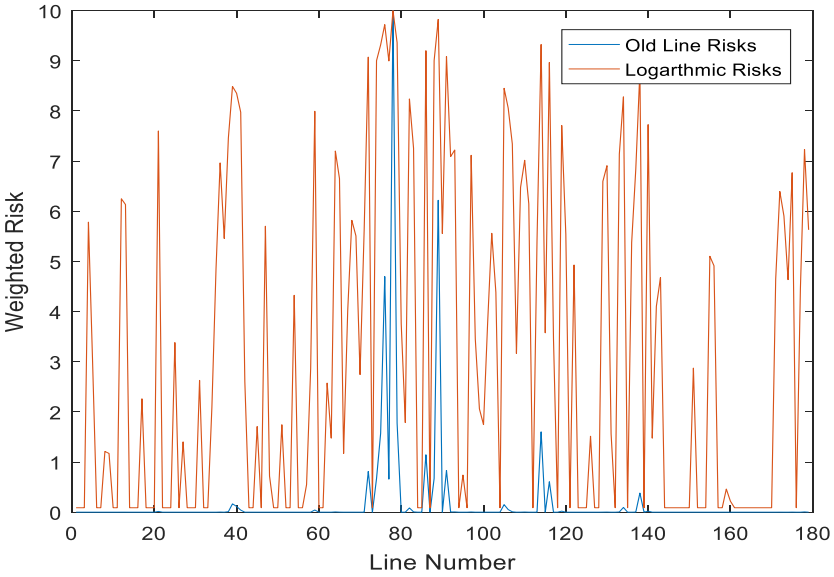

Figure 5-1: Line risks changed with risk weighting of 10

In addition, weights were added to the risk minimization and regularization terms to balance between moving towards a new operating point and staying at the current one. The objective function may be restated

as a function of risk-weighted line flows, $r(x)$, regularization, $l(x)$, and their associated weights, ω_r and ω_l, respectively.

$$\min_x (1-\alpha) f(x) + \alpha(\omega_r r(x) + \omega_l l(x)) \tag{5-23}$$

The weights in this new OPF are important as if they aren't picked well, the OPF may either stay at the least-cost generation operating point or move too far away from the operating point. A simple but seemingly effective method of choosing the weights is based on the Mean Squared Error (MSE) between MVA flows in the base case operating condition and the new operating condition. If the line flows are too different from the base case, the line risks will be less valid.

α was set to one to ignore the generation cost and then the weight ratio between regularization and risk minimization was found to keep the MSE below 10. On average, squared power flow on lines should not change by more than 10 pu which is equal to 1000 MW^2 or 31 MW. Once this ratio was found, α was set to 0.5 and the risk minimization weight was found by forcing the MSE below 1 pu (10 MW). The line flows and the MSE for the final result are shown in Figure 5-2 and Figure 5-3, respectively. These settings seemed to give good results for the overall dispatch.

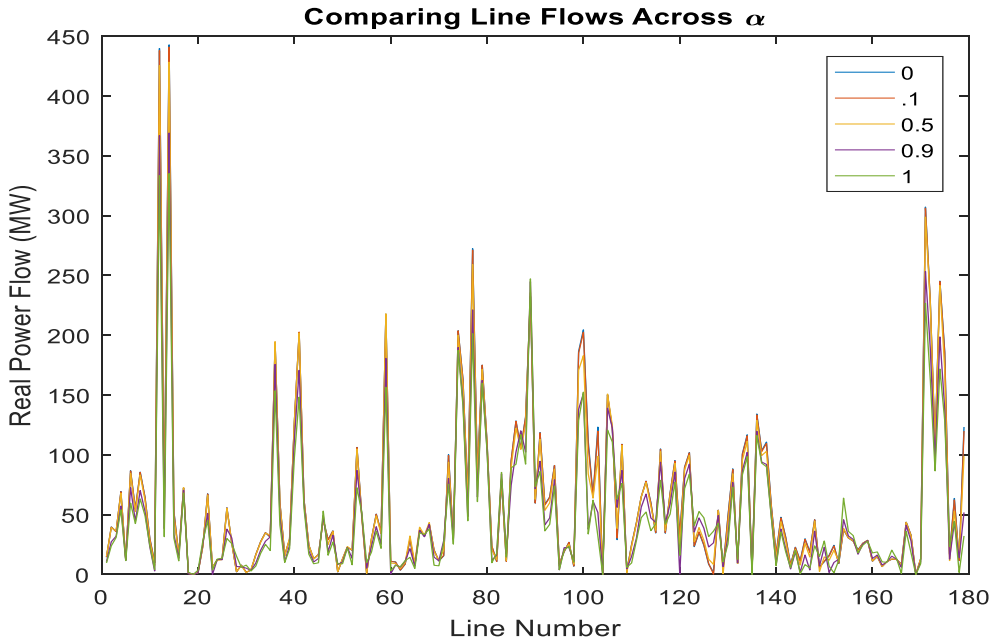

Figure 5-2: Line flows with a varying α

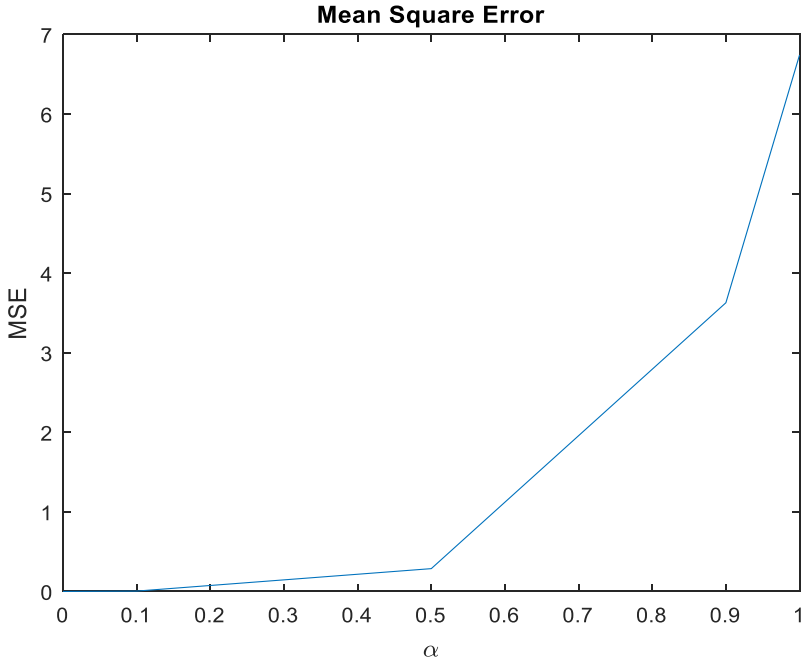

Figure 5-3: MSE with a varying α

5.3 Numerical Results

Using an α of 0.5 and $(\omega_r = 10, \omega_l = 100)$ the new substation risks of cascading is compared to the original dispatch's risks in Figure 5-4. The new dispatch reduces risk on all of the critical substations while sacrificing very little in risk on less critical substations. The risk from the loss of these critical substations were reflected in the most critical lines in the system. As such, the OPF could reduce the critical substation risk by reducing flow through the critical lines. As power flow was diverted from these lines towards other lines in the new dispatch, other substations gained more risk. However, the risk increase on these substations is neglible compared to the risk reduction of the most critical substations.

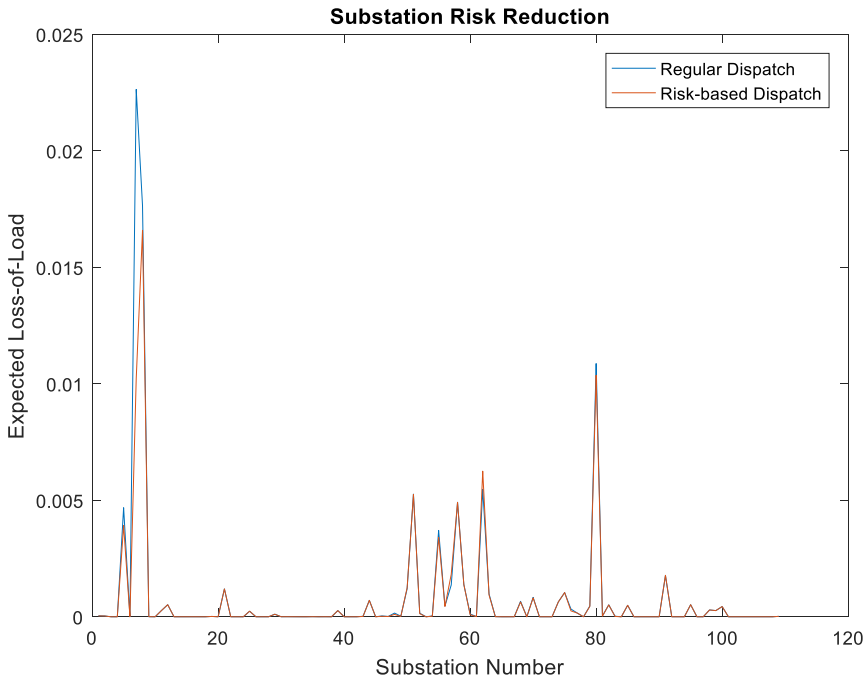

Figure 5-4: Expected loss-of-load from cascades for each dispatch

Figure 5-5 shows the result varying $\alpha = \{0, 0.01, 0.1, 0.5, 0.9, 1\}$ as a percent increase in system risk and cost. With the weights chosen (between risk minimization and regularization), the system risk can be reduced by as much as 40% with only a 3.5% increase in system cost. Further analysis may be done on the economics of the risk involved with each substation to do a system specific cost-benefit analysis to determine a good operating condition. It is ultimately up to the operator to decide how much tradeoff to accept between expected loss-of-load (risk) and system cost. It may be deemed appropriate to operate somewhere other than

$\alpha = 0$ on this curve during normal operating conditions or to simply operate there and move the operating point during abnormal conditions, e.g. hurricanes, tornados, etc.

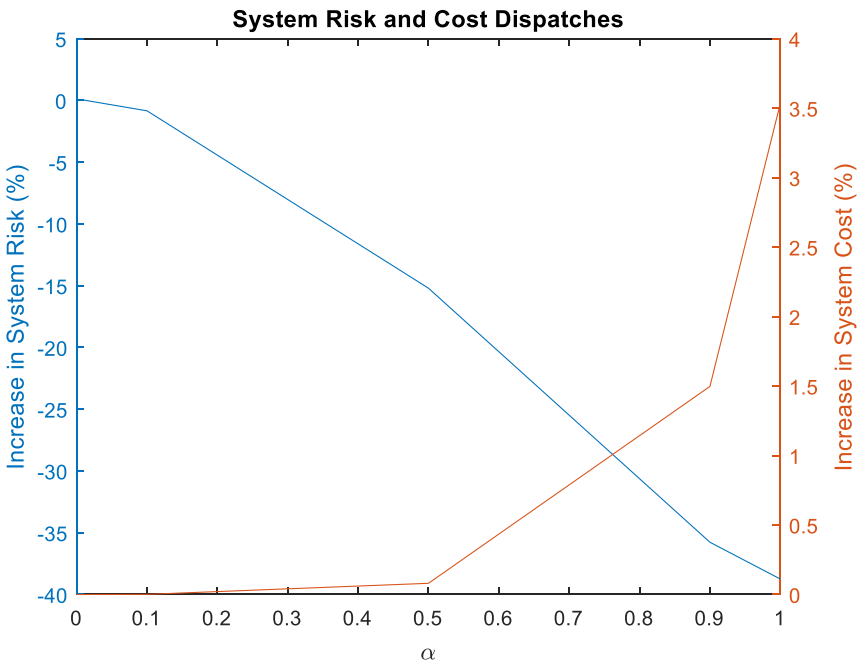

Figure 5-5: System Risk vs System Cost

This new OPF has weights involved that need to be tuned specific to circumstance and system. If poor weights are chosen, the new OPF may respond poorly and pick an operating point that does not efficiently mitigate system risk from cascades or may even make them work. The OPF was therefore tested with poor weighting to see how the new operating point would interact. The risks weight were set equal to 10 as before while the regularization term was set to one. α is set to one to ignore any effects the generation cost may have. This new OPF gives the current operating point $1/100^{th}$ of the weighting as the ones before allowing the OPF to move further from the current operating point.

Figure 5-6 and Figure 5-7 show the line flows and substations risks, respectively, resultant of inefficient weighting between risk minimization and regularization. Line flows differ drastically with many lines having much less flow. The overall system risk has still been reduced with the most critical substation risks being drastically reduced. However, the overall system risk has only been reduced by approximately 25% compared to a reduction of approximately 38% by a more efficiently weighted OPF. Compare this with an increase in system cost by almost 50% of the inefficient weighting versus the 3.5% of the more efficient weight. This new dispatch does less to decrease the overall expected loss-of-load from cascading at a much

higher cost economically. However, it still does reduce system risk by a non-negligible amount. This may suggest that the optimization is robust against different weights but has an optimal range for risk reduction.

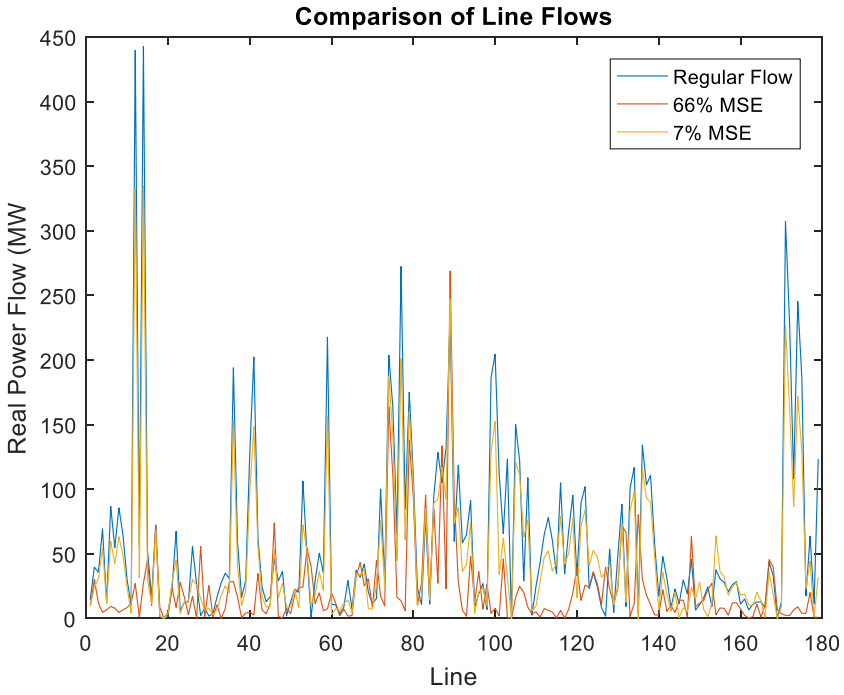

Figure 5-6: Comparison of line flows for well and poorly picked weights

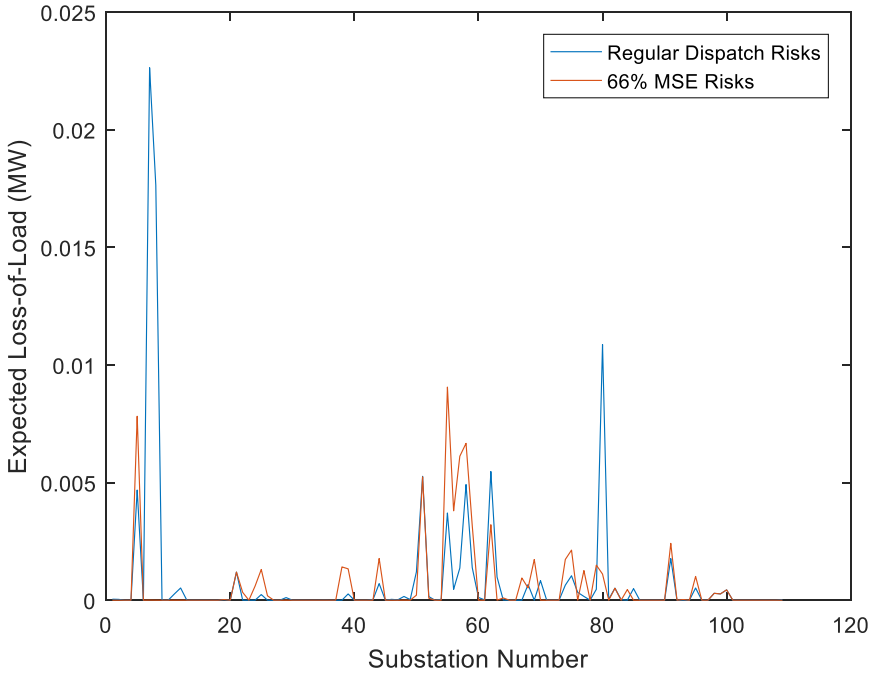

Figure 5-7: Comparison of substation risks for well and inefficient picked weights

6 CRITICAL SUBSTATION PROTECTION

In all the previous chapters, substation and system risk has only been viewed from the utilities' perspective. A big assumption in those chapters when looking at system risk is that the probability of losing any substation is uniform. This chapter explores those priors by framing the interaction of a potential attacker and the utility. A utility, under a constrained budget, cannot protect all of their critical substations and may benefit themselves by adding cheaper, less-rigid, or even fake protection that is indistinguishable from other protection to some substations to confuse a potential attacker on which substation are truly and actually protected.

6.1 Game Description and Formulation

In order to analyze this interaction, a zero-sum game is formulated. There are two players: one defender and one attacker. The defender is a utility whose goal is to protect substations to minimize system risk while the attacker is a malicious entity attacking substations to maximize system risk. The attacker is assumed to be a smart, rational (optimal) attacker that has obtained these risks, either through their own form of cascading analysis or through leaked, insider information. The defender moves first by adding real, fake or no protection to each substation. The attacker, seeing which substations have some form of protection, follows and decides which substations to attack. The attacker is able to distinguish protected substations from completely unprotected ones, but cannot differentiate between real and fake protection. Therefore, the attacker does not have complete information about the game state. The attacker is assumed to have no cost constraint. Rather, they are limited to the number of substations to attack.

Because the defender confuses the attacker by adding fake protection, this game is formulated as a dynamic, imperfect-information game. The game is dynamic because the defender and attacker make moves in a sequence allowing information to transfer between moves and is of imperfect information because the attacker cannot distinguish between true and fake protection. This gives the attacker game states that they can and cannot differentiate between. Utilities, or payoffs, to each player are assumed to be the risk associated with the loss of a particular substation. If the attacker attacks a substation with true protection, the attack is assumed to have failed and the payoff of the attack to both players is zero. Otherwise, if the attacker attacks a substation with fake or no protection, the payoff gain for the attack is equal to the risk of that substation for the attacker, and the defender loses that much in payoff. Once the attacker has taken an action, the game concludes.

This game is modeled as a dynamic, imperfect-information, zero-sum game in an extensive form game tree. An imperfect-information game in extensive form is concisely defined as a tuple $G = (N, A, H, Z, \chi, \rho, \tau, u, \mathcal{I})$ [95]. Here *N* is a set containing both players. *A* is a set containing specific actions at each node for both players. *H* and *Z* are sets of non-terminal and terminal nodes in the game tree, respectively. χ is the action function that maps nodes to actions. ρ is the player function that maps nodes to players. τ is the successor function that maps actions to new nodes. *u* is the utility function that maps terminal nodes to payoffs for each player. Finally, $\mathcal{I}$ is a set of "information sets" or partitions of nodes for each player such that the current player cannot differentiate between nodes within each partition.

A simple example of an extensive form game tree is shown in Figure 6-1 for a two substation system where the defender has the budget to protect one substation and fake protection on one substation or fake protection on both. Actions *B*-*C* represent the defender protecting or faking protection on the highest risk substation, actions *D*-*E* are analogous on the lowest risk substation, and actions *F*-*H* represent protecting one substation and faking the other (both combinations), or faking both. Once the defender has decide their protection scheme, the attacker then chooses to attack one substation as labeled by actions *a*, …, *h*. Each blue oval is an information set for player two where they cannot differentiate between nodes reached by different possible actions of the defender. Note that the attacker has the same attack options in an information set while having different attack possibilities in different information sets in order to model information gotten from the defender having gone.

For example, the defender's actions of *F* (protect substation 1 and fake 2), *G* (protect substation 2 and fake 1), and *H* (fake substation 1 and fake 2) lead to a system state with substation 1 and 2 both having forms of protection that player two cannot differentiate between. Yet the attacker knows both substations have forms of protection so his choices are different than if only one substation had protection. The number of these information sets depends on the number of substations as well as the budget the defender allocates to the game. The defender will always have one singleton information set. That is, it only has one node in its one information set. After the defender has built their defenses, the attacker then decides which of the two substations to attack. Both players will then be allotted payoffs corresponding to their actions as labeled by terminal leaves.

In this game, the defender's goal is to choose an optimal distribution $x \triangleq [x_1, x_2, \ldots., x_{\mathrm{M}}]^T \in \mathrm{X}$ over possible combinations of true and fake physical protection on substations where $X = \{x \in \mathbb{R}^M | x \geq 0, \sum_{m=1}^{M} x_m = 1\}$. This probability distribution can be seen as a randomization of the defender to confuse the

attacker about which substations have actual protection and which have fake or as a belief of the attacker on the defender's actions. Analogously, the attacker wants to choose a set of optimal distributions $y = \{y^{(1)}, y^{(2)}, \dots, y^{(K)}\}$ over all possible distinguishable protection scenarios where $y^{(k)} \triangleq [y_1, y_2, \dots., y_N]^T \in Y$ and $Y = \{y \in \mathbb{R}^N | y \geq 0, \sum_{n=1}^{N} y_n = 1\}$. This distribution can be thought of as the same. *M* is the number of combinations over which the defender can decide, K is the number of distinguishable game states for the attacker, and *N* is the number of substations considered by both players.

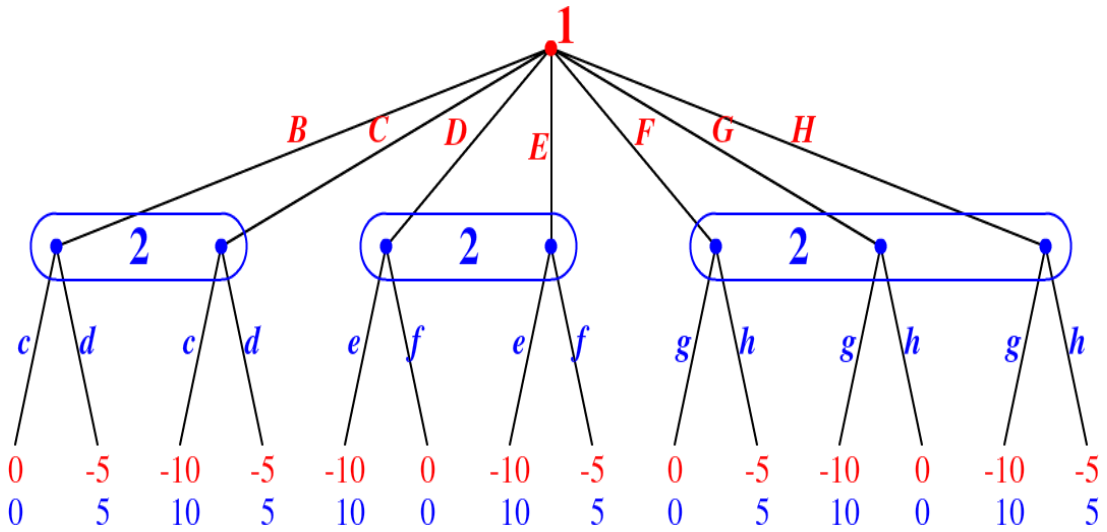

Figure 6-1: Two Substation Game Tree

Sequence game form is utilized to avoid an exponential game size increase in transforming dynamic games to strategic/normal form [96]. Instead of looking at action combinations, we consider sequences each player takes. Define S_i as the set of sequences of moves for player i:

$$S_i = \{\emptyset\} \cup \{\sigma_h \, a | \; h \in H_i, \qquad a \in A(h)\} \tag{6-1}$$

σ_h : sequence for player i to get to information set h
$\sigma_h a$: σ_h with action a taken afterwards
H_i : set of all information sets belonging to player i
A(h): set of all moves at information set h

This set includes the empty sequence where has not been able to go yet. We may then define a sparse (*N* x *M*) matrix payoff matrix *A* where a_{ij} is the payoff for the defender when they follow sequence *i*, and the attacker follows sequence *j*. *A* is a sparse matrix as each element in this matrix is zero unless it reaches a terminal node in the game tree where it equals the payoffs expressed above. As the empty sequence is

considered for both players, one of the columns and rows needs to be all zero to correspond to these sequences. The attacker's payoff matrix is defined as $-A$ as it is a zero-sum game.

Next, a realization plan, x, is defined that maps a sequence of moves to realization plan probability distribution vectors at each relevant information set for the defender (player one) [96]. Formally, $x : S_1 \rightarrow \mathbb{R}$ subject to

$$x(\sigma) \geq 0 \; \forall \; \sigma \in S_1 \tag{6-2}$$

$$x(\emptyset) = 1 \tag{6-3}$$

$$\sum_{a \in A(h)} x(\sigma_h a) = x(\sigma_h) \quad \forall \; h \in H_1 \tag{6-4}$$

Equation (6-2) and (6-4) forces the mapping onto a probability vector at each information set while equation (6-3) states that the probability of player *i* going through the empty set sequence is one. Note that the condition that all variables be less than or equal to one is implicit in this definition. An analogous mapping is defined as y for the attacker (player two). This mapping allows both a probability distribution for the defender as well as a set of probability distributions for the attacker. As this is a single-act game where each player only has one chance to move, these realization plans have a one-to-one mapping to behavioral strategies [95]. In matrix form this mapping is defined as

$$Fy = f, \qquad y \geq 0 \tag{6-5}$$

$$Ex = e, \qquad x \geq 0 \tag{6-6}$$

We may then define the expected system risk as $S_R = x^T A y$. Given that the defender (player one) wishes to minimize expected system risk while the attacker (player two) wishes to maximize it, the defender's formulation may be formed as a min-max problem given by equation (6-7)-(6-9).

$$\min_x \max_y x^T A y \tag{6-7}$$

s.t.

$$Ex = e, Fy = f \tag{6-8}$$

$$x \geq 0, y \geq 0 \tag{6-9}$$

This may be converted to a simple max/min problem by converting the optimization layers with their dual [96]. Consider a fixed strategy for the defender, the attacker then wants to maximize risk.

$$\max_{y}(x^T A)y \tag{6-10}$$

s.t. $$Fy = f \tag{6-11}$$

$$y \geq 0 \tag{6-12}$$

The dual of this is given by:

$$\min_{u} f^T u \tag{6-13}$$

s.t. $$F^T u - x^T A \geq 0 \tag{6-14}$$

Both these linear programs have feasible solutions; so by the theorem of strong duality, their optimal value is the same, $f^t u^* = (x^T A)y^*$. Therefore, the problem may be reformulated as a min-min problem or simply a minimization problem where the defender is trying to minimize $x^T Ay$ through a proxy $f^T u$ with their choice of *x*. Similar reasoning can find the realization plan (strategy) for the attacker. The min-max realization plan for the defender and the max-min realization plan for the attacker is given by equations (6-15)-(6-18) and

(6-19)-(6-22), respectively.

$$\min_{u,x} f^T u \tag{6-15}$$

subject to

$$Ex = e \tag{6-16}$$

$$F^T u - A^T x \geq 0 \tag{6-17}$$

$$x \geq 0 \tag{6-18}$$

$$\max_{v,y} e^T v \tag{6-19}$$

subject to

$$Fy = f \tag{6-20}$$

$$E^T v - Ay \leq 0 \tag{6-21}$$

$$y \geq 0 \tag{6-22}$$

This game has an equilibrium (x^*, y^*) with a value of $S_R^* = f^T v = -e^T u = x^{*T} Ay^*$. These maxmin plans, or security plans [97], consider the worst case scenario. The defender is trying to minimize the maximum risk they can allocate to the attacker while the attacker is trying to maximize the minimum damage they can inflict. These security strategies correspond to Nash equilibria (NE) in zero-sum games as any payoff that benefits one player hurts other player [97]. This game can have multiple NEs where all players are playing optimally. However, all zero-sum game NEs are saddle-point equilibria that result in the same expected system risk [97], namely the value of the game $S_R^* = x^{*T} Ay^*$. Furthermore, all equilibria are interchangeable. That is, if (x_1^*, y_1^*) and (x_2^*, y_2^*) are both NE, then so are (x_2^*, y_1^*) and (x_1^*, y_2^*). Any combination of equilibria strategies is also an equilibrium [97].

6.2 Game Reduction

While the number of nonzero entries in this LP is linear in the size of the game tree, the game tree for this game is exponential in the number of substations and becomes intractably large as more substations and larger budget are considered. Each substation has three discrete protection possibilities making the number of unique combinations of actions for the defender 3^N, where N is the number of substations considered while the attacker has 2^N information sets or distinguishable game states. As the attacker has to consider each substation for attack at each system state, distinguishable or not, the number of terminal nodes of the game tree assuming k attacks is $\binom{N}{k} * 3^N$. Fortunately, this game can be reduced because most defense strategies are not practical and will not be considered in any equilibrium.

In order to show this, we create a strategically equivalent sequential game. In this game, the defender first decides how many substations they would like to consider for any form of protection. Afterward, the defender decides which substations should have protection. This is strategically equivalent to the previous game because the defender intrinsically decides these two things while deciding their actions. The attacker and defender do not gain or lose any information or advantage by this reformulation. The example given above in Figure 6-1, transforms to Figure 6-2. For clarity, the original actions have been labeled the same as in Figure 1 with the new actions labeled appropriately. This now creates a game with different subtrees or proper subgames of the original game. A proper subgame in game theory is a subset of the original game with the following properties [98].

- It has a single decision node (non-terminal) that is the only node in its information set. This can be seen, for example, as the defender's node to decide between on the high or low risk substation in Figure 6-2.
- If a node is in a subgame, then all of its successors are as well.
- If an information set has a node in a subgame, then all of the nodes in the information set are in the same subgame. Subgames do not share different nodes of information sets.

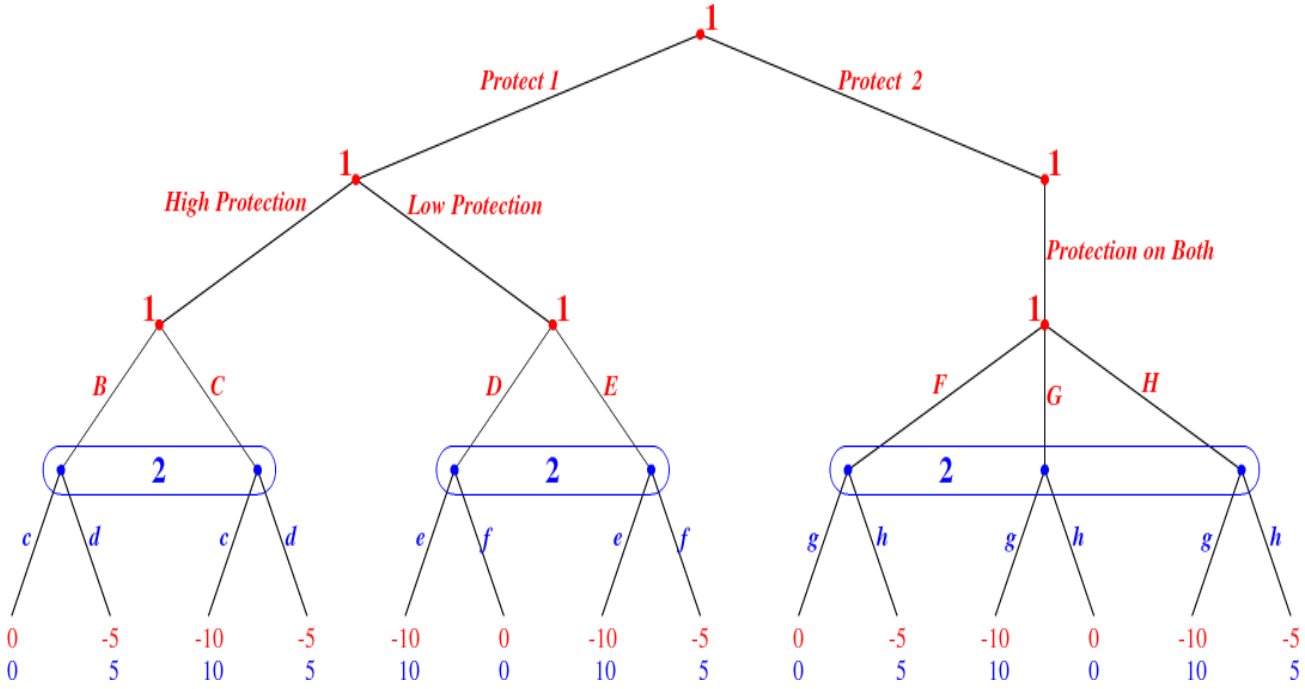

Figure 6-2: Modified two substation game tree

Thus, the new game in Figure 6-2 has six different subgames including the original game itself.

We then introduce the concept of Subgame Perfect Nash Equilibrium (SPNE). SPNE is a stronger concept of NE in that all SPNE are NE but not all NE are SPNE. A strategy profile that is an SPNE is a strategy profile that is also an NE at all subgames of the original game including the original game itself [99]. This stronger concept allows us to remove options that would not be played in an SPNE. This may and likely will remove many NE from the game. However, as SPNE represents a stronger concept to NE, and all NE admit the same value in zero-sum games, these removed NE are deemed irrelevant. The following proposition shows that in a subgame where *N* substations are considered for protection, the only NE strategy for the defender is to consider the top *N* risk substations.

Proposition: Consider a subgame of the original game where the defender decides to protect N substations, any strategy where the defender does not consider the top *N* risk substations for protection in the subgame will never be considered in an NE of the subgame.

Proof: Assume the substations in a system are ordered, 1, …, *M* based on their risk, high to low, such that $R_1 \geq \cdots \geq R_N \geq \cdots \geq R_M$. Let A be the set of n actions the defender takes to get to all nodes in an information set $\mathcal{I}$, and let $A(i)$ be an action in A to get to one node in $\mathcal{I}\ \forall\ i = 1, \ldots, n$. Define $\sigma_1(A(i))$ as

the probability the defender assigns to taking action $A(i)$ to get to the information set $\mathcal{I}$. Finally, define ρ_k to be the probability the attacker assigns to attacking substation *k* at information set $\mathcal{I}$.

The probability of faking a substation protection may be obtained through a payoff weighted sum of probabilities. The probability of truly protecting a substation *k* given that substation *k* is covered:

$$\Gamma_k = 1 - p_{fake}^{(k)} = 1 - \frac{\sum_{i=1}^{n} \sigma_1\big(A(i)\big) * U_k(i)}{R_k} \tag{15}$$

$$U_k(i) = \begin{cases} 0 & \text{if truly protected in sequence action } A(i) \\ R_k & \text{if faked in sequence action } A(i) \end{cases}$$

$\sigma_1\big(A(i)\big) * U_k(i)$ will be zero under three conditions: $U_k(i) = 0$, $\sigma_1\big(A(i)\big) = 0$ or both. These scenarios correspond to truly protecting the substation, considering the action to cover substation *k* with zero probability or both, respectively. Thus, $\sum_{i=1}^{n} \sigma_1\big(A(i)\big) * U_k(i)$ will only contain the risk weighted probabilities of faking substation k and can be normalized with R_k to obtain the probabilities. The expected payoff of any information set, $\mathcal{I}$, is then given as:

Assume that there is a Nash equilibrium strategy profile of this subgame (Γ^*, ρ^*) with an expected system risk of R^*, that has *N*-1 of the top *N* substations protected as well as another lower risk substation *p* in consideration for protection at information set $\mathcal{I}$. Let substation *k* be the one substation in the top *N* not in consideration, $k \leq N, p > N$. The expected payoff is then defined as:

$$R^* = \sum_{\substack{i=1, \\ i \neq k}}^{N} \rho_i^* (1 - \Gamma_i^*) R_i + \sum_{\substack{i=N+1 \\ , i \neq p}}^{M} \rho_i^* R_i + \rho_k^* R_k + \rho_p^* \big(1 - \Gamma_p^*\big) R_p \tag{17}$$

Γ_i: probability of truly protecting substation i
ρ_i: probability of attacking substation i
M: Number of substations in system

As $R_k > R_p$ by construction, it is to the attacker's benefit and the defender's detriment to attack substation *k* with increasing probability in order to increase system risk, i.e. $\rho_k^* > \rho_p^*$. In fact, the maximum occurs when all probability is given to substation *k* with no probability assigned to substation *p*.

Consider now a new strategy profile for the two players where the attacker plays the same strategy, and the defender's strategy is identical with the exception that $\Gamma_k' = \Gamma_p^*$ and $\Gamma_p' = \Gamma_k^* = 0$. The system risk under this new equilibrium is:

$$R' = \sum_{\substack{i=1,\\ i\neq k}}^{N} \rho_i^*(1-\Gamma_i^*)R_i + \sum_{\substack{i=N+1,\\ i\neq p}}^{M} \rho_i^* R_i + \rho_k^*(1-\Gamma_p^*)R_k + \rho_{\mathrm{p}}^* R_p \tag{18}$$

Let us then compare the difference in expected system risk.

$$R' - R^* = \Gamma_p^*\left(\rho_p^* R_p - \rho_k^* R_k\right) \tag{19}$$

As $\rho_k^* > \rho_p^*$ and $R_k > R_p$,

$$R' < R^* \tag{20}$$

This new strategy has lower risk than the supposed equilibrium strategy (Γ^*, ρ^*). The defender may unilaterally reduce the system risk by changing their strategy to considering the top *N* risk substations when considering *N* substations for protection. As a NE is defined informally as an equilibrium where no one player can do better by unilaterally changing their strategy, any strategy where the defender does not consider the top *N* risk substations when considering *N* substations for some form of protection is not in a Nash Equilibrium. ■

As the strategies where the defender does not consider the top *N*-risk substations correspond to subgames of the *N* substation subgame to the whole game, we may remove them, as they will not be played in an SPNE. Intuitively, this states that if a defender is playing against an attacker that knows the risk associated with each substation, it would be foolhardy to put protection on anything other than the top *N*-risk substations. Going back to the example in Figure 6-2, consider the subgame where the defender has decided that they would like to add some form of protection on one substation. This states that the defender should add protection to the highest risk substation with probability 1 (high protection subgame) and give no probability to the lowest risk substation (low protection subgame). Hence, we may remove the subgame that starts after the defender has decided to add protection to the low risk substation. Doing this to all

subgames will leave us with a game that has *K* subgames where *K* is equal to the maximum number of substations to add protection to as allowed by the budget of the defender.

Once the game is reduced by removing all these subgames, the game may be transformed back by removing the decisions of how many substations to protect and which substations to add that protection to. This will bring the game back to a game very similar to the original with many information sets missing. Thus allowing the sparse linear program to solve the new game to find the Nash Equilibrium.

6.3 Exploitation Versus Exploitability

A defender playing a NE strategy is only playing optimally under the assumption that the attacker is a fully rational player that is also playing optimally, i.e. both players are best-responding to each other [100]. If the attacker were to potentially play a non-optimal strategy, the defender could do no worse than against an optimal player with their NE strategy, but they would not be best-responding as the NE is a security strategy that assumes the opponent is the worst case opponent that will make no mistakes. This scenario is very unlikely to happen, as the opponent may not be fully aware of the substation risks due to outdated information, may simply not care, or a variety of other reasons. In this case, the defender could change strategies to consider many different types of opponents to exploit non-optimal attackers while preventing their new strategy from becoming too exploitable by a potential worst-case attacker.

The exploitability of a strategy for a player in a zero-sum game is easily defined as how far a player's expected payoff is from the value of the game under the NE [101]. For this game, the exploitability for player one and player two's strategy is how much the other player can increase and decrease, respectively, the expected system risk based on that strategy, i.e.

$$ex(\sigma_i) = \max_{\sigma_{-i} \in \Sigma_{-i}} (S_R^* - u_i(\sigma_i, \sigma_{-i})) \tag{6-23}$$

A strategy that can be exploited at most ϵ is considered ϵ-safe [101]. The set of ϵ-safe best responses against a strategy σ, $BR^{\epsilon-safe}(\sigma)$, is then defined as the set of all strategies that are ϵ-safe against any strategy σ. The defender would like to exploit any non-optimal attacker while making sure no intelligent attacker can exploit the defender's strategy by more than ϵ by picking their strategy from this set of best responses.

To model this, it is assumed the attacker has a probability (1-*p*) of being intelligent and fully rational and *p* of being non-optimal and attacking a substation according to some known prior distribution. This prior

probability distribution among substations may be built on certain beliefs the defender has on potential attackers. Attackers may have a preference of attack based on substation size, geographic location, distance from population, etc. In this way, the defender can come up with a strategy that balances utility against non-optimal attackers with security against worst case optimal attackers. If p is set to 0, the defender will play a robust NE strategy that has guarantees on minimum system risk against any player. As p approaches one, the defender will play a brittle best-response strategy against the player modeled by the pre-defined priors that will optimally punish mistakes due to the priors. This strategy will be far from robust against other strategies and can be taken advantage of.

To compute these optimal strategies, the game tree is configured, and a regret learning algorithm is used. A "chance" node based on p is added to the top of the game tree that splits it into two trees where the attacker plays optimally in one tree and uniformly in the other. Generally, in game theoretic frameworks this chance node is considered a "nature player" where nature decides whether the attacker will be optimal or non-optimal. Unfortunately, for this analysis, the game reduction assumptions no longer hold as the attacker is no longer considered rational and does not always pick the best strategy, and the LP quickly becomes intractable even in sparse representation. Counterfactual regret minimization is thus used in a repeated manner to approximate these equilibria. The defender and attacker both learn how to best respond under the assumptions of the attacker termed Restricted-Nash Response (RNR) in [101]. This allows the defender to exploit the attacker while remaining robust. Once this equilibrium is found, the attacker is then set to optimally best-respond to the defenders new strategy to find the exploitability of the defender's new strategy against a worst-case opponent.

6.4 CounterFactual Regret Minimization

Regret minimization employs repeated play between two players and minimizes their regret in each play through a process known as regret matching [102]. Regret minimization is proven to converge to a Nash Equilibrium for zero-sum two-player games and converges to a Course Correlated Equilibrium in general two player games [103]. Counterfactual regret minimization (CFR) is a powerful tool to approximate equilibria that stems from regret minimization for sequential games in order to reduce storage and computational burdens [103] and has been shown to be able to approximate solutions to games with as many as 10^{12} nodes [102]. Instead of minimizing regret over all actions in the entire action space, each player minimizes their counterfactual regret at each information set. Thus, players need only know the history of play as well as their previous payoffs but not necessarily the opponent's payoffs. Define the counterfactual value to player i under

the current strategy profile, σ, as their expected payoff given that they have reached information set I (expected payoff at information set I) [102].

$$v_i(I,\sigma) = \sum_{z \in Z_I} u_i(z)\pi^{\sigma}_{-i}(h)\pi^{\sigma}(h,z) \tag{6-24}$$

Z_I: set of all terminal nodes z branching from information set I
$\pi^{\sigma}_{-i}(h)$: probability of players other than i playing to node h
$\pi^{\sigma}(h,z)$: probability of getting to terminal node z from node h
$u_i(z)$: utility of terminal node z for player i
$\pi(h)$: probability of reaching h
$\pi(h,z)$: probability of reaching z given

For player i, at iteration t and each action in information set I, $a \in A(I)$, the counterfactual regret is the difference between the expected utility of that action and the counterfactual value given a strategy at information set I [102].

$$r_i^t(I,a) = v_i(I,\sigma^t_{I\to a}) - v_i(I,\sigma) \tag{6-25}$$

Here $\sigma^t_{I\to a}$ is identical to σ with the exception that player i plays action a at I with a probability of 1.0. This regret is a measure of how much player i would prefer to play action a over following the current strategy σ^t. These regrets are cumulated for each player over all game iterations to define their cumulative counterfactual regret [102].

$$R_i^T(I,a) = \sum_{t=1}^{T} r_i^t(I,a) \tag{6-26}$$

Using regret matching iteratively then guarantees that R_i^T/T is driven to zero. More formally, the average regret is bounded by (6-27) [102].

$$\frac{R_i^T}{T} < \frac{\Delta_i |\mathcal{I}_i| \sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}} \tag{6-27}$$

Δ_i is defined as the range of utilities for player *i*, $|\mathcal{I}_i|$ is the number of information sets for player *i*, $|A(\mathcal{I}_i)|$ is the maximum number of actions player *i* has at any of their playable nodes. Using this matching, we may define each iteration strategy profile as

$$\sigma_i^{T+1}(I,a) = \frac{R_i^{T+}(I,a)}{\sum_{b\in A(I)} R_i^{T+}(I,b)} \tag{6-28}$$

where $R_i^{T+} = \max(R_i^T(I,a),0)$ is the nonnegative cumulative regret [102]. If the nonnegative cumulative regret at information set *I* is zero, the current strategy is set to uniform among all actions at information set *I*. The procedure averages each players strategy over the iterations to get $\bar{\sigma}_i^T$. In zero-sum games, if $R_i^T \leq \epsilon$ for both players, then the average strategy profile $\bar{\sigma}^T = \{\bar{\sigma}_1^T, \sigma_2^T\}$ is a 2ϵ-NE. Note that an ϵ-NE is an approximation of a true equilibrium who's strategy satisfies:

$$u_i(\sigma) + \epsilon \geq \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i}) \quad \forall i \tag{6-29}$$

The iterative process may converge faster through a process of smart sampling. Instead of spanning the entire game tree every iteration, parts of the tree may be sampled through a process known as Monte Carlo Counterfactual Regret Minimization (MC-CFR). The approach avoids the whole tree while leaving the counterfactual regrets the same in expectation [104]. Using this approach the average regret, with a probability of 1-*p*, is bounded by [105].

$$R_i^T \leq \left(M_i + \frac{\sqrt{2|\mathcal{I}_i||\mathrm{B}_i|}}{\sqrt{p}}\right)\left(\frac{1}{\delta}\right)\left(\frac{\Delta_i\sqrt{|A(\mathcal{I}_i)|}}{\sqrt{T}}\right) \tag{6-30}$$

A version in this work called average strategy sampling [105] is used. For each traversal of the game tree, only a single action is sampled at each node not belonging to player *i* according to the current average strategy profile, $\bar{\sigma}_{-i}{}^T$. At any of player *i*'s nodes, each action is sampled with a probability of

$$\rho(I,a) = \max\left(\epsilon, \frac{\beta + \tau s_i^T(I,a)}{\beta + \Sigma_{b\in A(I)} s_i^T(I,b)}\right) \tag{6-31}$$

β is a parameter that tries to force the current strategy to a more uniform distribution to keep from making bad decisions when the current cumulative regret is not a good approximation of the equilibrium. τ makes sure that any action at iteration *t* is always sampled with a probability of at least $\frac{1}{\tau}$. Finally, ϵ makes sure all actions are always considered for sampling with a probability of ϵ. The thought behind this is that at the beginning, the sampling will sample all actions at player *i*'s nodes with uniform distribution as player *i* does not know which actions are better. As the player learns the better actions, gets a clearer idea of their regrets, and the cumulative regret grows; the algorithm starts trusting the player's choices more for sampling. In this way, we sample from all actions in the beginning and start to avoid wasting computation by sampling actions that will not be played in an actual equilibrium.

6.5 Numerical Results

Each substation is considered to have a specific cost for protection installation that is pre-calculated by the defender and may be a factor of geography, weather, etc. Lesser/fake protection cost should be correlated with the cost of true protection at a given substation, as many of the costs will be similar at a substation. For simplicity in this work, it is assumed the cost of adding true, rigid protection to any substation is equal, and the cost of faking protection on any substation is the same and 25% of the cost of rigid protection. To analyze multiple simultaneous attacks, the game tree was modified such that each action of the attacker no longer represented a substation attack but rather an attack on a subset of substations. Thus, at each information set the attacker had: *N* actions (one attack), $\binom{N}{2}$ actions for two attacks, and $\binom{N}{3}$ actions for three attacks where *N* is the number of substations.

6.5.1 Nash Equilibrium

Figure 6-3 overlays substation risks with the system risk for the first 10 substations. The bars concatenate the risks of the next *n* high risk substations. For example, bar one shows the risk of the most critical substation (blue), the second most critical (teal) and the third most critical substation (yellow). Assuming the risks are independent of one another and no substations are protected, bar one shows the system risk if the top substation were attacked (~0.2), the top two were attacked (~0.37) or the top three were attacked (~0.5). Bar two then shows the same with the highest risk substation being excluded. The line graphs overlaid show the overall expected system risk from the game if the top n substations were attacked. The top x-axis is normalized by the cost of adding true, rigid protection to a substation. A value of 1.0 means the defender has

the budget to add true protection to only one substation. A value of 0.25 means the defender has the budget to add fake protection to only one substation.

-

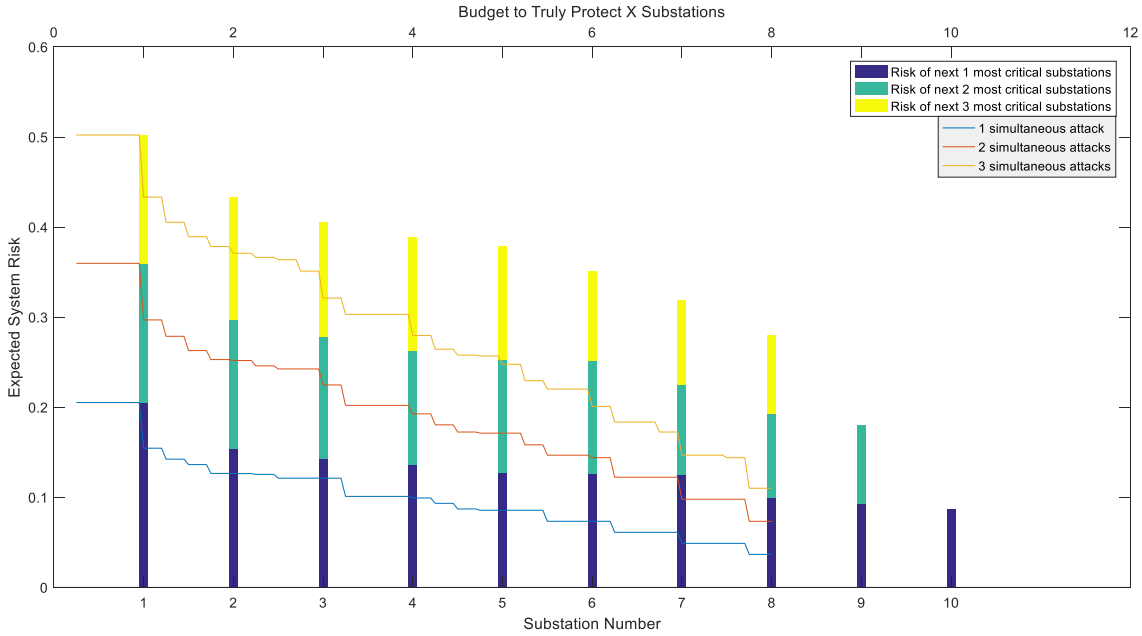

Figure 6-3: System Risk and Substation Risk

As is intuitive, the attacker gains the maximum amount in system risk if the defender does not have the budget to add rigid protection to one substation. As budget is allowed, the defender reduces the system risk more by confusing the attacker rather than adding rigid protection to the top risk substations. Consider a budget to be able to truly and rigidly protect four substations. If the defender only protects the top four risk substations, the system risk will be equal to the fifth highest substation in the risk rank (assuming one attack), approximately 0.13. However, with this new strategy, the system risk is equal to approximately 0.10. The defender has reduced the expected system risk by approximately 23% by adopting this new random strategy with an equivalent budget. While the game theory framework does account for multiple simultaneous attacks at once, one should be hesitant to use this with the previous cascading analysis. Assuming substation risks are independent from one another is a very strong assumption, and most likely incorrect given how the cascading analysis was done. However, if some form of joint substation loss analysis were to be done, the framework would allow it as it only changes the payoffs of each terminal leaf in the game tree.

The defender forces the attacker to consider lower risk, unprotected substations by adding more budget until the point where it is no longer beneficial on expectation for the attacker to attack unprotected substations.

At this point, the attacker randomizes over the top risk substations in such a way as to maximize their expected payoff from attack given their beliefs on which substations the defender actually protected. Consider the case where there is one attack on the system. The attacker starts attacking the first substation with a probability of one if the defender does not have enough money to defend. Once the defender has enough budget to add true protection, the attacker simply attacks substation two with a probability of one. The defender then includes enough budget to add a lesser/fake protection, and the attacker switches to substation two with a probability of one knowing that one of the substations is not truly protected. This continues on until the defender has the budget to add true protection to three substations. At a budget of 2.5, the defender has already forced the attacker to attack the 7th highest risk substation by randomizing one true protection and six fake protections in such a manner as to make any attack distribution have lower payoff in expectation than the 7th substation.

However, once the defender allocates another 0.25 in budget, the attacker can gain more payoff on expectation from randomizing over all seven protected substations since the decrease in expected payoff from the 7th to 8th substation is too much. At this point, the attacker believes that they can obtain more in payoff on expectation by guessing which substations are not truly protected than the 8th risk substation. The scenario of the attacker's switch from unprotected substations to randomizing is completely determined by the relative risks of the substations. If all substations have very similar risks, the attacker will tend towards unprotected substations instead of randomizing as can be seen in substations four to seven.

Constrained Analysis

To shed some insight into this behavior, different constraints are imposed on the defender. Figure 6-4 and Figure 6-5 shows the system risk and the probability of attack, respectively, while holding the total number of protected substations constant at 10 and varying the ratio of protected and unprotected substations. Figure 6-4 overlays fake exponential substation risks, $R_{SN} = 10e^{-\frac{SN}{3}}$, with the system risk to give a smoother idea of how much the protection is helping compared to the substation risks. In this scenario, all substations save one (the 11th) have some form of coverage. The defender then increases their "budget" to include more true protection to substations. At the beginning, the defender only has the budget to protect one substation so the defender has a high probability of guessing correctly and doing damage. As more substations get true protection, the chances of guessing correctly go down as well so the system risk decreases. This continues until the expected payoff to the attacker of guessing across the top 10 substations is less than the payoff of the 11th unprotected substation.

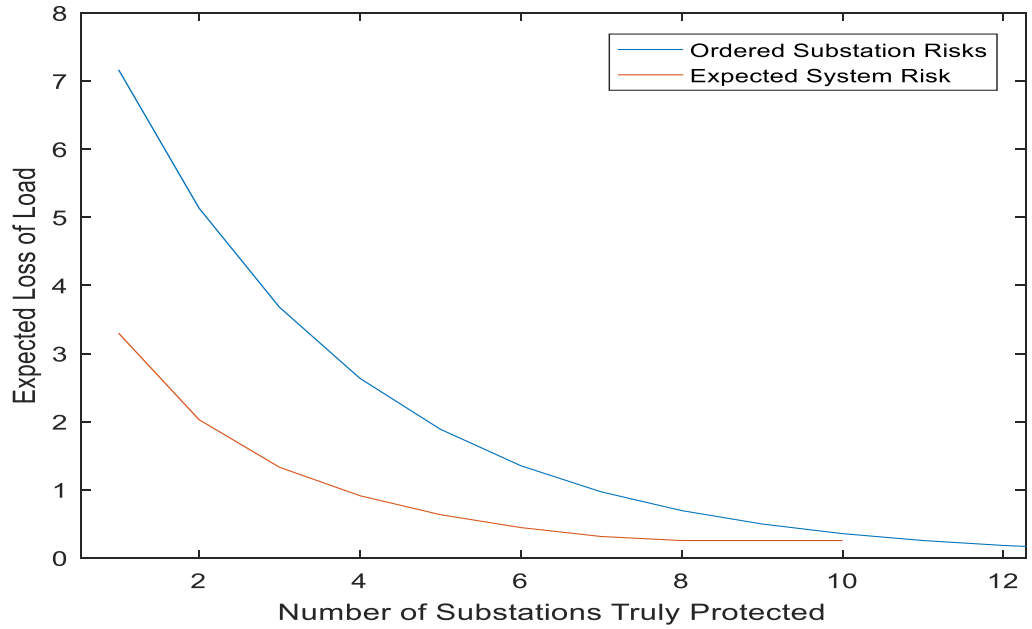

Figure 6-4: System Risk as more true protection is added

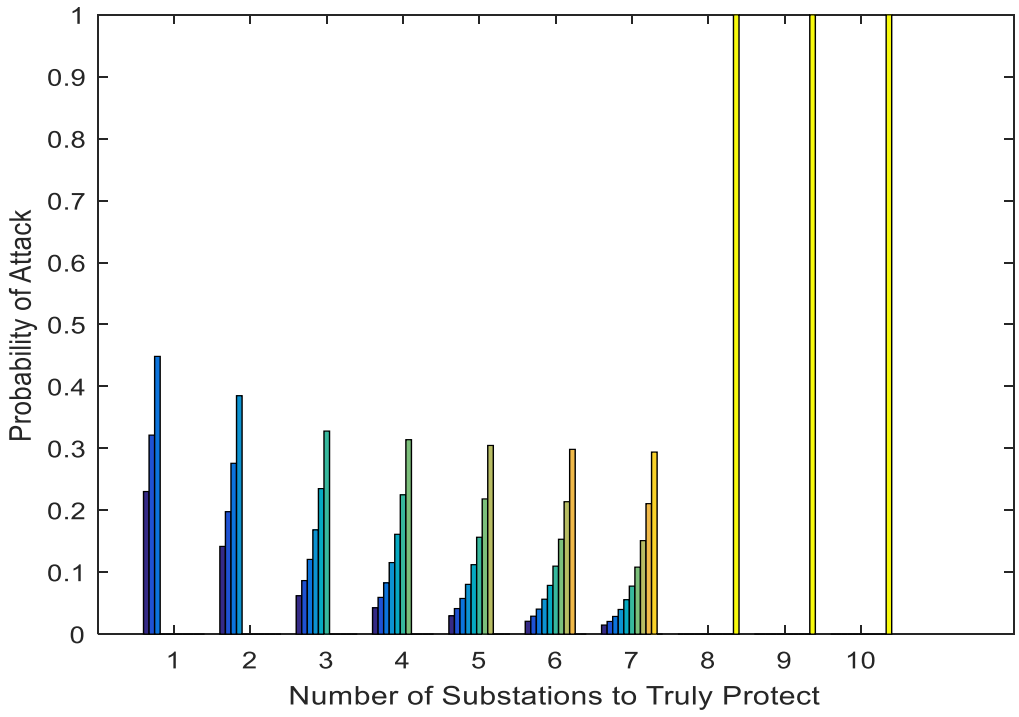

Figure 6-5: Attack probabilities as more true protection is added

Once the expected benefit of randomizing over protected substations is less than the risk of any unprotected substation, the attacker switches strategies to attacking only that substation and ignoring all

protected substations. The switch can be seen at eight truly protected substations where the expected system risk stops changing. Figure 6-5 illustrates the 10 different attack distributions across the top 11 substations as the aforementioned protections get added. The attacker will only randomize over the top three risk substations with one truly protected substation, the top four with two protections, and so forth. Overall, the attacker puts more weight on attacking the lower risk, protected substations as he believes that a rational defender would truly protect the higher risk substations. The attacker's distribution slowly weighs towards the lowest risk, protected substations until eight substations are truly protected at which point they attack the 11^{th} substation of risk. This switching of strategies for the attacker will happen quicker the larger the ratio of smallest risk substation to largest risk substation. Assuming the defender were to only protect 10 substations, they would only need allocate enough budget to truly protect eight.

The next constraint held the amount of true protection constant while allowing fake protection to vary. Figure 6-6 shows the result as the number of truly protected substations is kept constant at five while the number of fake protections is increased. The defender forces the attacker to randomize over more substations decreasing system risk further until three substation have fake protection. At this point, the expected system risk will remain constant regardless of how many fake protections are added. This is because the attacker does not keep considering more substations for attack as fake protection is increased beyond three. Rather, they randomize over the top ones since they know the number of true, rigid protection is kept at five. In this case, if the defender decides to protect five real substations, they stop decreasing expected system risk after the addition of three fake substation protections. In this way, the defender can determine how many fake protection schemes for other substations to consider for a given number of true protections. If the defender would like to reduce system risk further, they would need to consider more true/rigid protection.

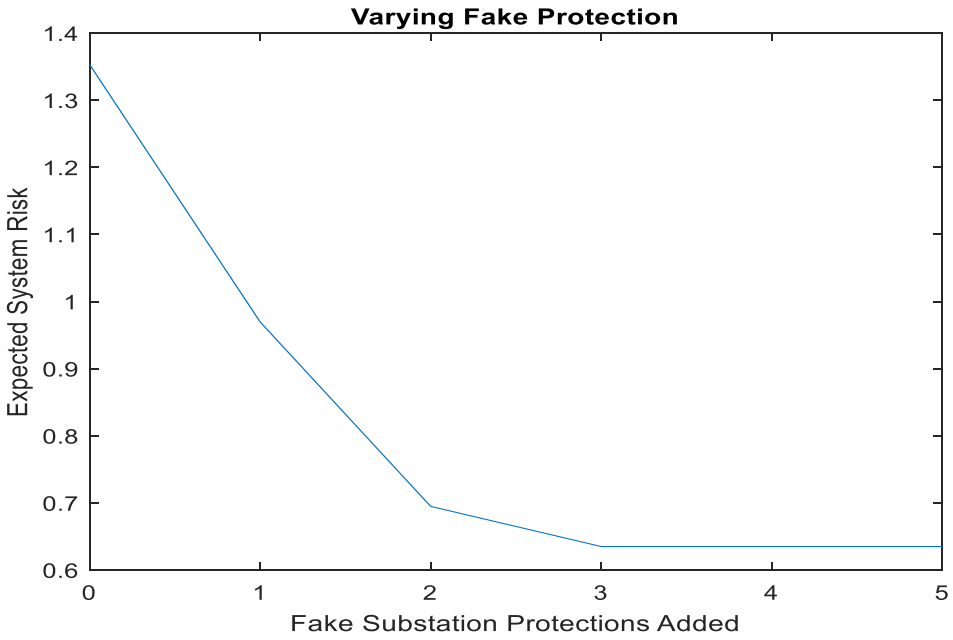

Figure 6-6: System Risk as fake protection is varied

Critical Substation Locations

The probability distribution across actions for the defender may be mapped to probabilities of protecting substations with strong, rigid (true) protection or cheaper, less rigid (fake) protection. As the defender will consider only the top N-risk substations for coverage, any substation that has any probability of having fake or true protection will have a probability of one of being covered. The results when the defender has a budget to add true protection to seven substations (or fake 28 substations) is shown in Figure 6-7. The critical substations are covered in boxes with the probability of true and fake protection shown as colored portions of the box. In this case, 13 substations have coverage: five substations are truly protected and eight are faked on expectation. This mapping to probabilities on substations is less useful than the action space as sampling on this distribution does not guarantee substations are covered with the cost allocated. However, it gives a general idea of how to protect important substations. The substations that tend to be critical are either connecting several areas together, have connection to many buses, or are heavily supporting the area.

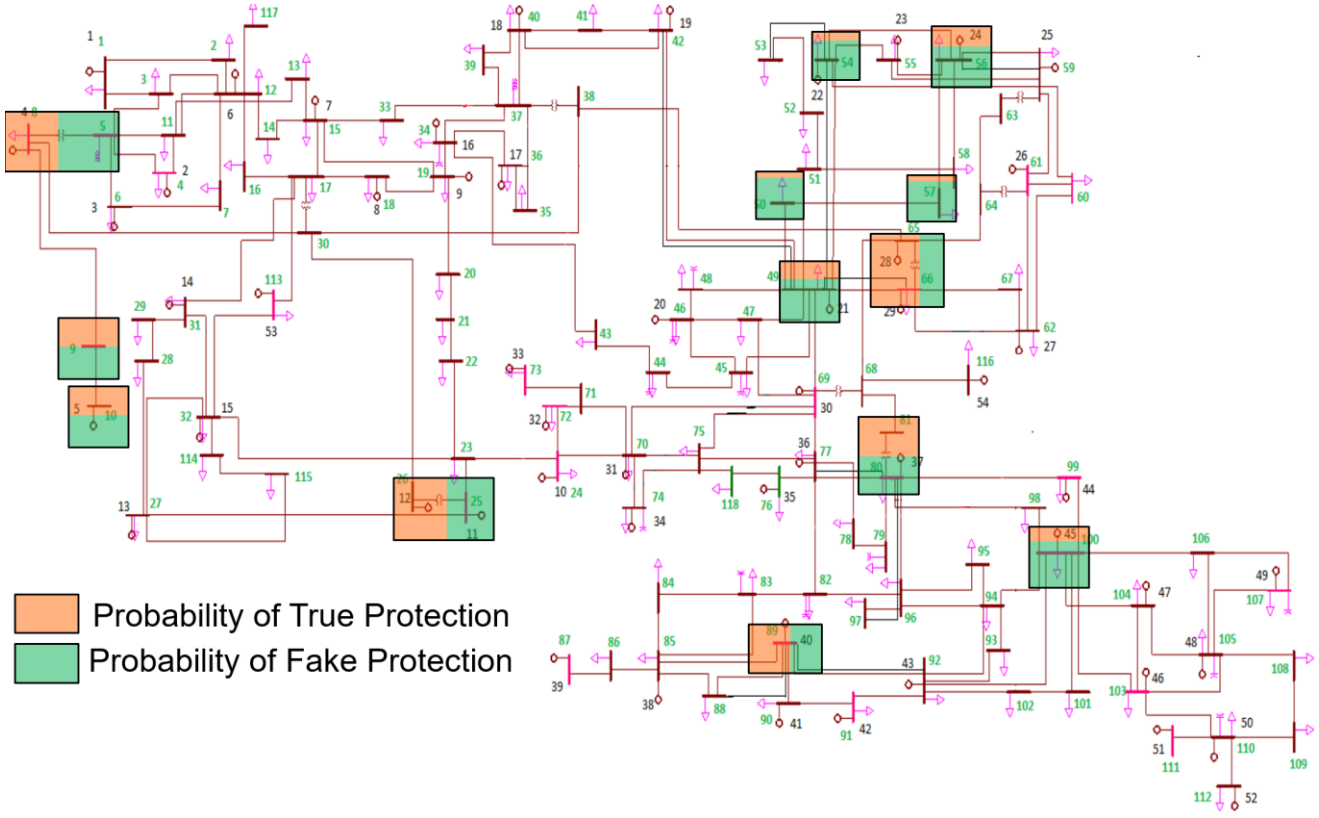

Figure 6-7: Probability of protecting critical substations

6.5.2 Exploitation of Unrational Attackers

For the study of exploitation versus exploitability, the NE is first calculated from the reduced game with a fixed budget. The probability that an attacker is non-optimal is then varied from 0 to 1 in increments of 0.1 for the defender to exploit against. The non-optimal attacker is modeled in three different forms: a one that does not care about the amount of damage done with a uniform distribution across substations, an attacker whose probability of attack has an increasing linear relationship with the substation risk rank and an attacker whose probability distribution is a decreasing linear function of substation risk rank. How much the defender's new strategy exploits this non-optimal attacker and how much the defenders new best-response strategy is exploited by an optimal attacker is recorded as a percentage difference from the NE and is shown in Figure 6-8. The MC-CFR algorithm was employed on two games: the new modified game with the full version of the optimal portion of the attacker and the new modified game with the reduced game of the optimal portion as discussed in the game reduction section. The full and reduced game results are shown on the left and right, respectively, in Figure 6-8.

Interestingly, the results of the MC-CFR on both the full and reduced games are very similar indicating that a reduced game may possibly be used to approximate results even though rationality assumptions on the attacker are not met. The defender stands to gain significant reduction in system risk against non-optimal attackers while losing very little to their worst case opponents regardless of the prior model. The defender gains the least from a decreasing probability non-optimal attacker model because it is the closest of the three models to the actual NE strategy (which looks similar to Figure 6-5). Depending on the attacker, the defender can reduce system risk anywhere from 5% to 23% by giving up only ~3% in system risk in the worst case against a tailored counterstrategy. Assuming attackers will have a 70% chance of being non-optimal is seemingly a good operating point for the defender as the graphs tend to saturate at that point with the defender starting to give up more risk to counterstrategies. This gives the defender an operating point that is robust, reduces the system risk significantly and increases system risk negligibly against an attacker that knows the defender has changed their strategy from their NE equivalent and creates an optimal counterstrategy.

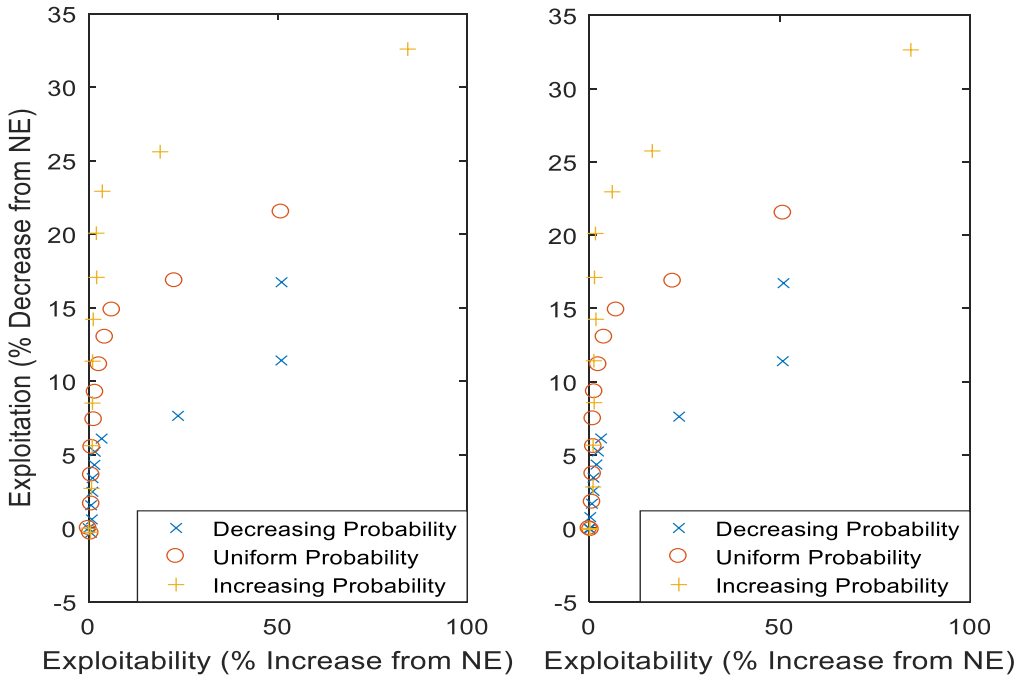

Figure 6-8: Defender's Exploitation vs. Exploitability

7 CONCLUSION AND FUTURE WORK

7.1 Conclusion

Substations are critical nodes in the system where many corridors of power flow meet, get transformed to different voltage levels, and rerouted to other areas of the power grid. They represent critical infrastructure in the electric power system whose loss may severely impact the reliability of the grid. This work has studied system risk associated with loss of substations. In doing so, it provides a methodology for ranking substations based on which are more critical to the reliable operation of the electric power system. A protection misoperation cascading model has been created to estimate the ranking of substation criticality with estimation of loss-of-load as a severity metric. With the combination of cascading probabilities gotten from the model, an overall view of system risk is obtained. Expected loss-of-load as a metric has the added benefit of showing potential economic impact due to cascading with the caveat that a hidden failure cascading model is an oversimplified model based on assumptions that may not give true expected loss-of-load. It has in the past, however, produced results that align with traditional large disturbance scenarios as seen in NERC data. Thus, it is believed that its resulting expected loss-of-load for each substation can provide a good ordering for substation criticality, and it is a good basis and starting point for any entity trying to make decisions based on criticality. The robustness of the ranking of substations has then been tested with different misoperation models as well as operating points.

The model has been further improved by including transient stability prediction for machines that may go unstable. As traditional transient stability approaches would require too much computational time for the combinatorial nature of cascades, a machine learning algorithm based on Random Forest of Decision Trees has been trained offline to predict whether machines will go unstable in cascading scenarios. The approach is a hybridization that uses time-domain simulations for the stable/unstable responses and load flow for the features. Many features, measured and derived from quasi-steady state analysis, were used for each generator based on intuition that these features give indication on stability. The skewness in the stable/unstable cases due to high power system reliability was overcome with an adaptive data synthetization technique where new unstable cases near classification boundaries are interpolated. The classifiers end up with very accurate results when considering stable or unstable cases. However, there is a tradeoff between the precision and recall of classifiers of which it is better for the classifier to be conservative with its classification of stable cases for system reliability and more liberal with its unstable classifications. This allows us to have more confidence that all the unstable cases were caught.

The comparison of these models showed the intuitive result that the more details that could go wrong in a model are included, the higher the estimated risk will tend to be, e.g. voltages collapsing in AC model vs DC model or machines going unstable in an AC model. However, it has also shown another interesting facet to cascading: system switching, loading shedding and redispatching can help a system lose less load on expectation. This was seen by comparing the models against their counterparts that did not have those features. Voltage collapse considered in the AC model caused an isolation in two substation losses that did not occur in the DC model and lessened the system risk of those two substations in the AC model compared to the DC. Transient instability caused generators to re-dispatch due to lack of power balance and also helped improve system risk compared to the AC model. These results in combination leads one to consider operations during a cascade may be usable as a tool for mitigating system failure.

Line risks gotten from cascading analysis were then used to create a risk mitigating OPF dispatch point. The risk-weighted line flows through all lines are added as a cost in the cost function to reduce power flowing through the lines that are involved in the cascading scenarios with the largest expected loss-of-load. A weighting parameter is added to allow an operator to consider the weighting between least-cost generation and least-risk dispatch. A regularization term is added to the least-risk dispatch to maintain the new dispatch point close to the original as line risks and risks in general are a function of the dispatch point and not constant. As the dispatch moves away from the original dispatch, line risks change; the OPF will no longer truly be minimizing risk. Mean square error of line flows from their original dispatch is used as a metric to determine weights between regularization and risk minimization. Many of the highest risk substations can have their risk substantially reduced without a large cost increase or increasing the risk of other substations.

Finally, the work studied the protection of substations by utilities when considering strong, rigid protection on some substations and lesser protection on other substations in a game-theoretic framework. The problem has been modeled as a sequential, zero-sum game, proved to be reducible to a smaller game, and solved by linear programming. System risk and potential attacker strategy has been studied varying the amount of each type of protection. A smart, rational attacker will tend to randomize their strategies over the protected substations if they believe that the unprotected substations' risk are too low. The defender's strategies have also been studied under non-optimality of attackers to allow a utility to decide how much preference to give to strategies against worst-case, intelligent, rational attackers versus strategies against ordinary malicious attacks. Assuming the attacker has a high probability of being non-optimal, a utility can stand to gain a lot from decrease of expected loss-of-load by moving away from a Nash equilibrium strategy while losing little against the very rare rational attacker. The non-optimal attacker in this work was studied as

having three prior distributions: uniform, directly proportional to the risk of the substation, and inversely proportional to the risk of the substation. While these were the models chosen for this work, it is trivial to change to a model that has an actual preference.

7.2 Future Work

Future work will focus in a few different aspects:

Probabilistic Load Flow incorporation: In this work, loads and generation are seen as constant and deterministic with randomness in the protection systems. Variance in generation and loads be incorporated by incorporating a probabilistic load flow.

Include dispatching as a method for attacker confusion: In this work, it is assumed the defending entity may only confuse attackers through the use of fake attackers. However, results from the risk dispatching may be used to confuse the attacker against the risks as well. Interestingly, even if we use a high MSE for power flows, we can still significantly reduce system risk. Thus, we can use these line risks as quasi-constant and create new dispatches with new substation risks. Hence, we can change the "game" in game theory to include OPF dispatches that the attacker cannot see. In this way, the defender may confuse the attacker about the actual risks of each substation potentially further reducing system risk.

Neural network training for system re-configuration: System re-configuration has been known to be able to reduce traditional congestion. There may be strategies to reconfigure the system during a cascade to reduce expected loss-of-load. This very complicated, non-linear problem would be very dependent on model of cascade used. However, there may be a way to train a Neural Network to figure out these reconfigurations through Reinforcement Learning methods. This may be taken a step further and a smart islanding technique may be implemented to reduce system risk from cascading.

Decreasing transient stability prediction time: While random forests, reduce the computational burden of transient stability prediction, they are still slow in comparison to other machine learning algorithms to prediction due to the number of trees that features need to go through. There may be a further way to reduce the computational burden by creating a simpler prediction model that is even more heavily skewed towards instability to the point where it has 100% recall at the cost of terrible precision. However, if this model then predicts stable, then the random forests need not do predictions. Otherwise, stability prediction will go through the random forest again.

Testing on a real, larger power system: This cascading model can be quite computationally complex. In a real power system, model reduction may have to be done before this model may be feasibly be implemented. This will also be subject to future work.

REFERENCES

[1] NERC, *Transmission System Planning Performance Requirements,* TPL-001-4, 2015.

[2] "Electric Disturbance Events (OE-417) Annual Summaries," Office of Electricity Delivery & Energy Reliability, 2017. [Online]. Available: https://www.oe.netl.doe.gov/OE417_annual_summary.aspx.

[3] P. C. o. E. Advisers, "Economic Benefits of Increasing Electric Grid Resilience to Weather Outage," Washington D.C., 2013.

[4] A. Atputharajah and T. Saha, "Power System Blackouts - Literature Review," in *International Conference on Industrial and Information Systems (ICCIS),* Sri Lanka, 2009.

[5] U.S.-Canada Power System Outage Task Force, "Final Report on the August 14, 2003 Blackout in the United States and Canad: Causes and Recommendations," DoE, 2004.

[6] K. Tweed, "Bulletproofing the grid [News]," *IEEE Spectrum,* vol. 51, no. 5, pp. 13-14, May 2014.

[7] U.S. Attorney's Office, "Woodring Sentenced to 15 Years for Attacks on Central Arkansas Power Grid," Office of Public Affairs, 18 June 2015. [Online]. Available: https://www.fbi.gov/contact-us/field-offices/littlerock/news/press-releases/woodring-sentenced-to-15-years-for-attacks-on-central-arkansas-power-grid.

[8] E-ISAC, "Analysis of the Cyber Attack on the Ukrainian Power Grid," Washington, DC, 2016.

[9] B. News, "Ukraine power cut 'was cyber-attack'," BBC, 2017 January 11. [Online]. Available: http://www.bbc.com/news/technology-38573074.

[10] NERC, *Critical Infrastructure Protection,* CIP 0-14-1, 2014.

[11] K. Tweed, "Attack on Nine Substations Could Take Down U.S. Grid," *IEEE Spectrum: Technology, Engineering, and Science News,* 13 March 2014.

[12] J. D. McDonald, Electric Power Substations Engineering, Third Edition, Boca Raton: CRC Press, 2012.

[13] D. E. d. l. Garza, "Hidden Failures in Protection Systems and its Impact on Power System Wide-area Disturbances," Blacksburg, 2000.

[14] J. Thorp, A. Phadke, S. Horowitz and S. Tamronglak, "Anatomy of Power System Disturbances: Importance Sampling," *International Journal of Electrical Power & Energy Systems,* vol. 20, pp. 147-152, 1998.

[15] R. Billington and P. Vohra, "Station initiated outages in composite system adequacy evaluation," *IEE Proceedings C - Generation, Transmission and Distribution,* vol. 134, no. 1, pp. 10-16, 1987.

[16] R. Allan and J. Ochoa, "Modeling and Assessment of Station Originated Outage For Composite System Reliability Evaluation," *IEEE Trasactions on Power Systems,* vol. 3, no. 1, pp. 158-165, 1988.

[17] J. Satish and R. Billinton, "Minimum Cost Analysis of Station Configurations," *IEEE Transactions on Power Delivery,* vol. 10, no. 4, pp. 1799-1805, 1995.

[18] W. Li and J. Lu, "Risk Evaluation of Combinative Transmission Network An Substation Configuration And Its Application In Substation Planning," *IEEE Transactions on Power Systems,* vol. 20, no. 2, pp. 1144-1150, 2005.

[19] R. N. Allan and A. N. Adraktas, "Terminal Effects and Protection System Failures in Composite System Reliability Evaluation," *IEEE Transactions on Power Apparatus and Systems,* Vols. PAS-101, no. 12, pp. 4557 - 4562, 1982.

[20] A. M. L. d. Silva, A. Violin, C. Ferreira and Z. S. Machado, "Probabilistic Evaluation of Substation Criticality Based on Static and Dynamic System Performances," *IEEE Transactions on Power Systems,* vol. 29, no. 3, pp. 1410-1418, 2014.

[21] A. M. L. d. Silva, J. L. Jardim, L. R. d. Lima and Z. S. Machado, "A Method for Ranking Critical Nodes in Power Networks Including Load Uncertainties," *IEEE Transactions on Power Systems,* vol. 31, no. 2, pp. 1341-1348, 2016.

[22] A. Torres and G. J. Anders, "Strategic Lines and Substations in an Electric Power Network," in *Innovations in Power Systems Reliability,* New York, Springer, 2011, pp. 169-190.

[23] D. T. Nguyen, Y. Shen and M. T. Thai, "Detecting Critical Nodes in Interdependent Power Networks for Vulnerability Assessment," *IEEE Transactions on Smart Grids,* vol. 4, no. 1, pp. 151-159, March 2013.

[24] I. Dobson, B. A. Carreras, V. E. Lynch and D. E. Newman, "Complex Systems Analysis of Series of Blackouts: Cascading Failure, Critical Points, and Self-Organization," *Chaos An Interdisciplinary Journal of Nonlinear Sciences,* vol. 17, 2007.

[25] S. Mei, X. Zhang and M. Cao, Power Grid Complexity, New York: Springer, 2011.

[26] J. Chen, J. S. Thorp and M. Parashar, "Analysis of Electric Power System Disturbance Data," in *Proceedings of the 34th Hawaii International Conference on System Sciences,* Maui, 2001.

[27] K. Bae and J. Thorp, "An Importance Sampling Application: 179 Bus WSCC System Under Voltage Based Hidden Failures and Relay Misoperations," in *Proceedings of the Thirty-First Hawaii International Conference on System Sciences,* Kohala Coast, 1998.

[28] K. Bae and J. Thorp, "A Stochastic Study of Hidden Failures in Power System Protection," *Decision Support Systems,* vol. 24, pp. 259-268, 1999.

[29] H. Wang and J. Thorp, "Optimal Locations for Protection System Enhancement: A Simulation of Cascading Outages," *IEEE Transactions on Power Delivery,* vol. 16, no. 4, pp. 528-533, 2001.

[30] J. Chen and J. Thorp, "A Reliability Study of Transmission System Protection via a Hidden Failure DC Load Flow Model," in *Fifth International Conference on Power System Management and Control,* 2002, 2002.

[31] J. Chen, J. Thorp and I. Dobson, "Cascading Dynamics and Mitigation Assessment in Power System Disturbances via a Hidden Failure Model," *International Journal of Electrical Power & Energy Systems,* vol. 27, no. 4, pp. 318-326, 2005.

[32] H. Mori, "State-of-the-Art Overview on Data Mining in Power Systems," in *Power Engineering Society General Meeting,* Montreal, 2006.

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

[34] L. Wehenkel and M. Pavella, "Decision tree approach to power systems security," *International Journal of Electrical Power and Energy Systems,* vol. 15, no. 1, pp. 13-36, December 1993.

[35] L. Wehenkel, M. Pavella, E. Euxibie and B. Heilbronn, "Decision Tree Based Transient Stability Method A Case Study," *IEEE Transactions on Power Systems,* vol. 9, no. 1, pp. 459-469, 1994.

[36] S. Rovnyak, S. Kretsinge, J. T. and D. Brown, "Decision Trees for Real-Time Transient Stability Prediction," *IEEE Transactions on Power Systems,* vol. 9, no. 3, pp. 1417-1426, August 1994.

[37] N. Hatziargyriou, S. Papathanassiou and M. Papadopoulos, "Decision trees for fast security assessment of autonomous power systems with a large penetration from renewables," *IEEE Transactions on Energy Conversion,* vol. 10, no. 2, pp. 315-325, 1995.

[38] B. C. Csáji, "Approximation with Artificial Neural Networks," Faculty of Sciences Eötvös Loránd University, Hungary, 2001.

[39] D. J. Sobajic and Y. H. Pao, "Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems," *IEEE Transactions on Power Systems,* vol. 4, no. 1, pp. 220-228, 1989.

[40] M. Djukanovic, D. J. Sobajic and Y.-H. Pao, "Neural-Net Based Unstable Machine Identification Using Individual Energy Functions," *International Journal of Electrical Power & Energy Systems,* vol. 13, no. 5, pp. 255-262, 1991.

[41] A. Edwards, K. Chan, R. Dunn and A. Daniels, "Transient Stability Screening Using Artificial Neural Networks Within A Dynamic Security Assessment System," *IEE Proceedings - Generation, Transmission and Distribution,* vol. 143, no. 2, pp. 129-134, 1996.

[42] E. V. Y. Mansour and M. A. El-Sharkawi, "Dynamic Security Contingency Screening and Ranking Using Neural Networks," *IEEE Transactions on Neural Networks,* vol. 8, no. 1, pp. 942-950, 1997.

[43] C.-W. Liu, M.-C. Su, S.-S. Tsay and Y.-J. Wang, "Application of a Novel Fuzzy Neural Network to Real-Time Transient Stability Swings Prediction Based on Synchronized Phasor Measurements," *IEEE Transactions on Power Systems,* vol. 14, no. 2, pp. 685-692, 1999.

[44] A. Schmilovici, "Support Vector Machines," in *The Data Mining and Knowledge Discovery Handbook,* New York, Springer, 2005, pp. 257-276.

[45] L. Moulin, A. d. Silva, M. El-Sharkawi and R. Marks, "Support Vector Machines for Transient Stability Analysis of Large-Scale Power Systems," *IEEE Transactions on Power Systems,* vol. 19, no. 2, pp. 818-825, 2004.

[46] D. You, K. Wang, L. Ye, J. Wu and R. Huang, "Transient Stability Assessment of Power System Using Support Vector Machine With Generator Combinatorial Trajectories Inputs," *International Journal of Electrical Power & Energy Systems,* vol. 44, no. 1, pp. 318-325, 2013.

[47] Y. Zhou, J. Wu, Z. Yu, L. Ji and L. Hao, "A Hierarchical Method for Transient Stability Prediction of Power Systems Using the Confidence of a SVM-Based Ensemble Classifier," *Energies,* 2016.

[48] M. B. Cain, R. P. O'Neill and A. Castillo, "History of Optimal Power Flow and Formulations," FERC, Washington D.C., 2012.

[49] Q. Wang, A. YANG and F. WEN, "Risk-based security-constrained economic dispatch in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 1, no. 2, pp. 142-149, 2013.

[50] X. Guo and J. McCalley, "Risk-Based Constraint Relaxation for Security Constrained Economic Dispatch," in *North American Power Symposium*, Charlotte, 2015.

[51] L. Roald, M. Vrakopoulou, F. Oldewurtel and G. Andersson, "Risk-based optimal power flow with probabilistic guarantees," *ScienceDirect*, vol. 72, pp. 66-74, 2015.

[52] W. Fu, "Risk Assessment and Optimization for Electric Power Systems," Iowa State University, Ames, 2000.

[53] J. C. Smith and C. Lim, "Algorithms for Network Interdiction and Foritification Games," in *Pareto Optimality, Game Theory, and Equilibria*, New York, Springer, 2008, pp. 609-644.

[54] J. C. Smith, M. Prince and J. Geunes, "Modern Network Interdiction Problems and Algorithms," in *Handbook of Combinatorial Optimization*, Springer, 2013, pp. 1949-1987.

[55] J. M. Lopez-Lezama, J. Cortina-Gomez and N. Munoz-Galeano, "Assessment of the Electric Grid Interdiction Problem using a nonlinear modeling approach," *Electric Power Systems Research*, vol. 144, pp. 243-254, 2017.

[56] J. Salmeron, K. Wood and R. Baldick, "Analysis of Electric Grid Security Under Terrorist Threat," *IEEE TRANSACTIONS ON POWER SYSTEMS*, vol. 19, no. 2, pp. 905-912, 2004.

[57] N. Romero, N. Xu, L. K. Nozick, I. Dobson and D. Jones, "Investment Planning for Electric Power Systems under Terrorist Threat," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 108-116, 2012.

[58] A. Delgadillo, J. M. Arroyo and N. Alguacil, "Analysis of Electric Grid Interdiction With Line Switching," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 633-641, 2010.

[59] D. Elizondo, J. d. L. Ree, A. Phadke and S. Horowitz, "Hidden Failures in Protection Systems and Their Impact on Wide-Area Disturbances," in *Power Engineering Society Winter Meeting*, Columbus, 2001.

[60] S. Tamronglak, "Analysis of Power System Disturbances due to Relay Hidden Failures," Blacksburg, 1994.

[61] Protection System Misoperations Task Force, "Misoperations Report," NERC, Atlanta, 2013.

[62] North American Electric Reliability Council, "Disturbance Analysis Working Group Database".

[63] S. H. Horowitz and A. Phadke, Power System Relaying, 3rd ed., John Wiley & Sons, 2008.

[64] V. Kristof and M. Mester, "Loss of excitation of synchronous generator," *Journal of ELECTRICAL ENGINEERING*, vol. 68, no. 1, pp. 54-60, 2017.

[65] R. C. Scharlach and J. Young, *Lessons Learned From Generator Event Reports*, Brooklyn Center, Minnesota, 2010.

[66] A. Wood and B. Wollenberg, Power Generation, Operation, and Control, 2nd ed., John Wiley & Sons, 2012.

[67] The Joint Board for the PJM/MISO Region, "Report on Security Constrained Economic Dispatch," FEDERAL ENERGY REGULATORY COMMISSION, Washington D.C., 2006.

[68] J. Thorp and A. Nagavi, "Load-Flow Fractals Draw Clues to Erractic Behaviour," *IEEE Computer Applications in Power,* vol. 10, no. 1, pp. 59-62, 1997.

[69] M. L. Crow, Computational Methods for Electric Power Systems, Boca Raton, Florida: Taylor & Francis Group, 2010.

[70] J. A. Bucklew, Introduction to Rare Event Simulation, New York: Springer, 2004.

[71] R. D. Zimmerman, C. E. Murillo-Sancchez and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning and Analysis Tools for Power Systems Research," *IEEE Transactions on Power Systems,* vol. 26, no. 1, pp. 12-19, Feb 2011.

[72] V. Balamourougan, T. Sidhu and M. Sachdev, "Technique For Online Prediction of Voltage Collapse," *IEE Proceedings - Generation, Transmission and Distribution,* vol. 51, no. 4, pp. 453-460, 2004.

[73] P. Kundur, N. J. Balu and M. G. Lauby, Power System Stability and Control, 7th ed., New York: McGraw-Hill Companies, 1994.

[74] D. J. Glover, M. S. Sarma and T. J. Overbye, Power System Analysis and Design, 5th ed., Stanford, CT: CENGAGE Learning Custom Publishing, 2011.

[75] A. R. Messina and V. Vittal, "Extraction of Dynamic Patterns from Wide-Area Measurements Using Empirical Orthogonal Functions," *Ieee Transactions on Smart Grid,* vol. 22, no. 2, pp. 682-692, May 2007.

[76] H. He, Y. Bai, E. A. Garcia and S. Li, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning," in *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Hong Kong, 2008.

[77] C. Elkan, "The Foundations of Cost-Sensitive Learning," in *Proc. Int. Joint Conf. Artificial Intelligence (IJCAI'01)*, 2001.

[78] G. M. Weiss, K. McCarthy and B. Zabar, "Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?," in *DMIN 7*, 2007.

[79] C. Drummond and R. Holte, "C4.5, Class Imbalance, and Cost Sensitivity: Why Under-Sampling Beats Oversampling," in *Proc. ICML '03 Workshop on Learning from Imbalanced Data Sets*, 2003.

[80] N. V. Chawla, L. O. Hall, K. W. Bowyer and W. P. Kegelmever, "SMOTE: Synthetic Minority Oversampling Technique," *Journal of Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[81] R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, John Wiley & Sons, Inc, 2001.

[82] A. Cutler, D. R. Cutler and J. R. Stevens, "Random Forests," in *Ensemble Machine Learning*, New York, Springer, 2012, pp. 157-176.

[83] L. Hyafil and R. L. Rivest, "Constructing Optimal Binary Decision Trees is NP-Complete," *Information Processing Letters,* vol. 5, no. 1, pp. 15-17, 1976.

[84] L. Rokach and O. Maimon, "Decision Trees," in *The Data Mining and Knowledge Discovery Handbook*, Springer, 2010, pp. 165-192.

[85] L. Breiman, "Random Forests," in *Machine Learning*, Netherlands, Kluwer Academic Publishers, 2001, pp. 5-32.

[86] R. Polikar, "Ensemble Learning," in *Ensemble Machine Learning*, New York, Springer, 2012, pp. 1-34.

[87] M. Sewell, "Ensemble Learning," UCL Department of Science, 2011.

[88] M. R. Segal, "Machine Learning Benchmarks and Random Forest Regression," in *Center for Bioinformatics & Molecular Biostatistics*, 2004.

[89] D. Barber, Bayesian Reasoning and Machine Learning, New York: Cambridge University Press, 2015.

[90] W. Y. Loh and Y. Shih, "Split Selection Methods for Classification Trees," *Statistica Sinica,* vol. 7, pp. 815-840, 1997.

[91] L. Breiman, J. Friedman, R. Olshen and C. Stone, Classification and Regression Trees, Boca Raton, Florida: CRCPress, 1984.

[92] W. Y. Loh, "Regression Tress with Unbiased Variable Selection and Interaction Detection," *Statistica Sinica,* vol. 12, pp. 361-386, 2002.

[93] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies,* vol. 2, no. 1, pp. 37-63, 2001.

[94] P. Bertail, S. J. Clémençcon and N. Vayatis, "On Bootstrapping the ROC Curve," *Advances in Neural Information Processing Systems*, 2009.

[95] Y. Shoham and K. Leyton-Brown, "Games with Sequential Actions: Reasoning and Computing with the Extensive Form," in *Multiagent Systems: Alorithmic, Game-Theoretic, and Logical Foundations*, 2009, pp. 118-130.

[96] N. Nison and T. Roughgarden, Algorithmic Game Theory, New York: Cambridge University Press, 2007.

[97] T. Basar and G. J. Olsder, Dynamic Noncooperative Game Theory, New York: Academic Press, 1995.

[98] D. Fudenberg and J. Triole, Game Theory, Cambridge: The MIT Press, 1991.

[99] K. Leyton-Brown and Y. Shoham, Essentials of Game Theory A Concise, Multidisciplinary Introduction, Morgan & Claypool, 2008.

[100] M. Ponsen, S. d. Jong and M. Lanctot, "Computing Approximate Nash Equilibria and Robust Best-Responses Using Sampling," *Journal of Artificial Intelligence Research,* vol. 42, pp. 575-605, 2011.

[101] M. Johanson and M. Zinkevich, "Computing Robust Counter-Strategies," *Advances in neural information processing systems,* pp. 721-728, 2008.

[102] M. J. Martin Zinkevich, "Regret Minimization in Games with Incomplete Information," *NIPS,* pp. 1729-1736, 2007.

[103] R. Gibson, "Regret Minimization in Non-Zero-Sum Games with Applications to Building Champion Multiplayer Computer Poker Agents," *arXiv:1305.0034,* 2013.

[104] M. Lanctot, K. Waugh, M. Zinkevich and M. Bowling, "Monte Carlo Sampling for Regret Minimization in Extensive Games," *Advances in Neural Information Processing Systems,* pp. 1078-1086, 2009.

[105] N. Burch, M. Lanctot, D. Szafron and R. G. Gibson, "Efficient Monte Carlo Counterfactual Regret Minimization in Games with Many Player Actions.," *Advances in Neural Information Processing Systems,* pp. 1880-1888, 2012.

[106] National Research Council, Making the Nation Safer. The Role of Science and Technology in Countering Terrorism, Washington, D.C.: National Academy Press, 2002.

[107] National Research Council, "Physical Security Considerations for Electric Power Systems," in *Terrorism and the Electric Power Delivery System*, Washington D.C., The National Academies Press, 2012, pp. 32-37.

[108] D. Bienstock and S. Mattia, "Using mixed-integer programming to solve power grid blackout problems," *Discrete Optimization,* vol. 4, no. 1, pp. 115-141, March 2007.

[109] E. Bompard, C. Gao, R. Napoli, M. Masera and A. Stefanini, "Information Impact on the Risk Analysis of the Malicious Attack against Power," in *iREP Symposium- Bulk Power System Dynamics and Control*, Charleston, 2007.

[110] J. Salmeron, K. Wood and R. Baldick, "Worst-Case Interdiction Analysis of Large-Scale Electric Power Grids," *IEEE Transactions on Power Systems,* vol. 24, no. 1, pp. 96-104, February 2009.

[111] J. M. Arroyo and F. D. Galiana, "On the Solution of the Bilevel Programming Formulation of the Terrorist Threat Problem," *IEEE Transactions on Power Systems,* vol. 20, no. 2, pp. 789-797, May 2005.

[112] L. Mili, Q. Qiu and A. Phadke, "Risk Assessment of Catastrophic Failures in Electric Power Systems," *International Journal of Critical Infrastructures,* vol. 1, no. 1, pp. 38-63, 2004.

[113] K. R. Apt, *Strategic Games,* 2011.

[114] R. Gibson, "Computing Strong Game-Theoretic Strategies and Exploiting Suboptimal Opponents in Large Games," Pittsburgh, 2015.

[115] T. V. Cutsem and C. Vournas, Voltage Stability of Electric Power Systems, New York: Springer-Verlag, 2007.

[116] M. Moghavvemi and F. Omar, "Technique for Contingency Monitoring and Voltage Collapse Prediction," *IEE Proceedings - Generation, Transmission and Distribution,* vol. 145, no. 6, pp. 634-640, 1998.

[117] B. A. Carreras, V. Lynch, I. Dobson and D. Newman, "Critical Points and Transitions in an Electric Power Transmission Model for Cascading Failure Blackouts," *Chaos: An Interdisciplinary Journal of Nonlinear Science,* vol. 12, no. 4, pp. 985-998, 2002.

[118] D. Nedic, I. Dobson, D. Kirschen, B. A. Carreras and V. Lynch, "Criticality in a Cascading Failure Blackout Model," *Electrical Power and Energy Systems,* vol. 28, no. 9, pp. 627-633, 2006.

[119] H. T. Ma, M. L. Crow, B. H. Chowdhury and A. Lininger, "Cascading Line Outage Prevention with Multiple UPFCs," in *29th North American Power Symposium (NAPS '07)*, 2007.

[120] 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,* vol. 31, no. 3, pp. 2085-2095, May 2016.

[121] P. W. Sauer and M. A. Pai, Power System Dynamics and Stability, Prentice Hall Publications, 1997.

[122] L. Breiman, "Bias, Variance, and Arcing Classifiers," 1996.

[123] D. Fabozzi and T. V. Cutsem, "Simplified Time-Domain Simulation of Detailed Long-Term Dynamic Models," in *Power Energy Society General Meeting*, 2009.

[124] K. Xiao, C. Zhu, W. Zhang, X. Wei and S. Hu, "Stackelberg Network Interdiction Game: Nodal Model and Algorithm," in *2014 5th International Conference on Game Theory for Networks (GAMENETS)*, Beijing, 2014.

[125] H. Sreekumaran, A. R. Hota, A. L. Liu, N. A. Uhan and S. Sundaram, "Multi-Agent Decentralized Network Interdiction Games," 2015.

[126] M. A. Pai, Energy Function Analysis for Power System Stability, New York: Springer, 1989.

[127] H.-D. Chiang, Direct Methods for Stability Analysis of Electric Power Systems: Theoretical Foundation, BCU Methodologies, Applications, Hoboken, New Jersey: John Wiley & Sons, Inc., 2010.

[128] IEEE PES CAMS Task Force, "Initial Review of Methods for Cascading Failure Analysis in Electric Power Transmission Systems," in *IEEE Power Engineering Society General Meeting*, Pittsburgh, 2008.

[129] C. Zheng, V. Malbasa and M. Kezunovic, "Regression tree for stability margin prediction using synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1978-1987, 2012.

[130] W. Li, Risk Assessment of Power Systems: Models, Methods, and Applications, Piscataway: IEEE Press, 2005.

APPENDIX A – BASE OPERATING POINT

Generator Profile		
Bus	PG	VG
1	2.73E-06	1.075413
4	2.21E-06	1.095133
6	2.40E-06	1.089559
8	2.19E-06	1.07644
10	314.102	1.090037
12	65.58335	1.087814
15	2.49E-06	1.087848
18	2.47E-06	1.088073
19	2.51E-06	1.087214
24	2.40E-06	1.083074
25	153.092	1.1
26	220.1611	1.065256
27	2.47E-06	1.085232
31	5.567226	1.078942
32	2.56E-06	1.082632
34	2.38E-06	1.096268
36	2.41E-06	1.09133
40	2.87E-06	1.082947
42	2.90E-06	1.084161
46	15.27041	1.078443
49	159.3461	1.099417
54	44.30836	1.1
55	6.75E-06	1.091271
56	8.19E-06	1.085664
59	113.3109	1.1
61	111.6159	1.089354
62	1.89E-06	1.085323
65	265.8789	1.050699
66	265.6573	1.1
69	341.3538	1.1
70	2.42E-06	1.076021
72	3.02E-06	1.06685
73	2.78E-06	1.068212
74	2.51E-06	1.068654
76	2.63E-06	1.06358
77	1.83E-06	1.089837
80	310.1007	1.1
85	1.47E-06	1.08721
87	2.481821	1.099999
89	331.5677	1.1
90	1.38E-06	1.094721
91	1.48E-06	1.090588
92	1.52E-06	1.092902
99	1.63E-06	1.095746
100	161.9721	1.097785
103	26.11562	1.095486
104	1.74E-06	1.088411
105	1.76E-06	1.088956
107	1.83E-06	1.083841
110	1.66E-06	1.094455
111	23.28747	1.099193
112	1.66E-06	1.094431
113	2.34E-06	1.091828
116	1.70E-06	1.04412

Loading Profile					
Bus	PD	QD	Bus	PD	QD
1	38.25	20.25	60	29.25	13.5
2	15	6.75	61	0	0
3	29.25	7.5	62	21	5.25
4	22.5	9	63	0	0
5	0	0	64	0	0
6	39	16.5	65	0	0
7	14.25	1.5	66	49.5	15
8	52.5	17.25	67	51	20.25
9	0	0	68	0	0
10	0	0	69	0	0
11	35.25	7.5	70	35.25	8.25
12	25.5	12	71	0	0
13	10.5	0.75	72	51	27
14	67.5	22.5	73	45.75	21
15	18.75	7.5	74	53.25	19.5
16	8.25	2.25	75	29.25	24
17	45	25.5	76	97.5	19.5
18	33.75	18.75	77	40.5	20.25
19	13.5	2.25	78	15	7.5
20	10.5	6	79	8.25	5.25
21	7.5	3.75	80	18	11.25
22	5.25	2.25	81	0	0
23	46.5	9.75	82	15.75	7.5
24	12.75	5.25	83	36	7.5
25	0	0	84	58.5	31.5
26	0	0	85	48.75	7.5
27	18	3	86	9	5.25
28	32.25	20.25	87	0	0
29	44.25	17.25	88	22.5	12
30	0	0	89	0	0
31	17.25	6.75	90	31.5	23.25
32	44.25	19.5	91	28.5	11.25
33	24.75	6.75	92	11.25	6.75
34	23.25	12.75	93	25.5	6
35	20.25	8.25	94	27.75	13.5
36	15	17.25	95	16.5	11.25
37	0	0	96	3.75	2.25
38	0	0	97	17.25	12
39	27.75	7.5	98	28.5	18.75
40	27.75	17.25	99	23.25	19.5
41	13.5	5.25	100	32.25	12
42	12	6	101	21	9
43	39.75	16.5	102	1.5	0.75
44	21	7.5	103	6	2.25
45	25.5	0	104	29.25	22.5
46	15	8.25	105	18.75	9.75
47	65.25	22.5	106	6	2.25
48	12.75	3	107	16.5	5.25
49	12.75	6	108	15	6
50	13.5	3.75	109	24.75	11.25
51	17.25	8.25	110	0	0
52	84.75	24	111	0	0
53	47.25	16.5	112	0	0
54	63	13.5	113	0	0
55	9	2.25	114	0	0
56	9	2.25	115	0	0
57	277	113	116	0	0
58	58.5	2.25	117	0	0
59	57.75	10.5	118	0	0

APPENDIX B – CODE

Min/Max Linear Program

```
function [x, y, fcn_val] = getRealizationPlans(PayoffMatrix, InfoSets,
player_two_choice_size)
% This function takes in the payoffmatrix for a single-act, imperfect-
information,
% zero-sum game.
%
% fcn_val = x^T * PayoffMatrix * y
%
% Inputs:
%
% PayoffMatrix: (N1 x N2) Payoff matrix using sequence forms. a_ij
corresponds to the
% payoff of the gametree if player one plays sequence i and player two
% plays sequence j. N1/N2 sequences for player one/two
%
% InfoSets: This is a (Nd x 1) vector that numbers the information set
each
% of player one's sequences go to for player two.
%
% player_two_choice_size: How many choices player two is considering for
% attack. This implicitly states that player two considers the same
amount
% of substations for attack at each information set.
%
% Outputs:
%
% x: (N1 x 1) vector behavorial strategy for the defender.
%
% y: (N2 x 1) vector of different behavorial strategies at different
% information sets. Number of behavioral strategies =
% N2/player_two_choice_size. Basically this gives a probability vector
for
% player two at each of their information sets.
%
% fcn_val: Value of the game at equilibrium.

if size(PayoffMatrix,2) == max(InfoSets)*player_two_choice_size
    PayoffMatrix = [zeros(size(PayoffMatrix,1),1) PayoffMatrix];
    PayoffMatrix = [zeros(1, size(PayoffMatrix,2)); PayoffMatrix];
    InfoSets = [0; InfoSets];
elseif size(PayoffMatrix,2) ~= max(InfoSets)*player_two_choice_size+1
    error('Something went wrong with matrix size');
else
    InfoSets = [0; InfoSets];
end

Amat = -sparse(PayoffMatrix);
Infos = unique(InfoSets);

% Set up the programmer.
opt = optimoptions('linprog');
opt.Display = 'none';
opt.Algorithm = 'dual-simplex';

best_response_data = [];
% Making sure the stuff is proper.
if isempty(best_response_data)
    best_response_data = struct();
    best_response_data.flag_1 = false;
    best_response_data.flag_2 = false;
end
%% Algorithm Game Theory Pg 73
%% Constraint Matrices
% Player one
% Create the constraint coefficient matrix
E = sparse([1 zeros(1,size(PayoffMatrix,1)-1);
    -1 ones(1,size(PayoffMatrix,1)-1)]);

% Create the constraint equality matrix
e = sparse([1; 0]);

% Player two
% Initialize the constraint coefficient matrix
F = sparse([1 zeros(1,size(PayoffMatrix,2)-1);
    -1*ones(length(Infos)-1,1) ones(length(Infos)-1,
size(PayoffMatrix,2)-1)]);

% Create the constraint equality matrix
f =  sparse([1; zeros(length(Infos)-1,1)]);

% Modify the constraint coefficient matrix
t = size(F,1)-1;
m = player_two_choice_size;
if m ~= 1
    if size(Amat,2) ~= player_two_choice_size + 1
        for ii = 1:t
            F(ii+1,setdiff(2:size(F,2), (1+m*(ii-1))+1:m*ii+1)) = 0;
        end
    end
end

%% Player 2 Formulation
% Equality Matrices
Aeqy = sparse([zeros(size(F,1),size(e,1)) F]);
beqy = f;

% Inequality Matrices
Ay = sparse([-E.' Amat]);
by = zeros(size(Ay,1),1);

% Objective function coefficient matrix
f_p2 = sparse([e; zeros(size(F,2),1)]);

% Lower bound vector
lby = zeros(size(f_p2));
lby(1:size(E,1))  = -Inf;
y = linprog(f_p2,Ay,by,Aeqy,beqy,lby,[],[],opt);

% Get rid of the unconstrained variables as well as the realization
% probability of the null set (which will be 1).
y_realization = y(size(E,1)+2:end);

%% Player 1 Formulation
% Equality Matrices
Aeqx = [zeros(size(E,1), size(f,1)) E];
beqx = e;

% Inequality Matrices
Ax = sparse([F.' -Amat.']);
bx = sparse(zeros(size(Ax,1),1));

% Lower bound vectors
lbx = zeros(size(Ax,2),1);
lbx(1:size(f,1)) = -Inf;

f_p1 = [f; zeros(size(E,2),1)];

% Linprog minimizes f, this problem requires that we max, so make f neg
x = linprog(-f_p1,Ax,bx,Aeqx,beqx,lbx,[],[],opt);

% Get rid of the unconstrained variables as well as the realization
% probability of the null set (which will be 1).
x_realization = x(length(x)-size(E,2)+2:end);

fcn_val = -y(1);
y = y_realization;
x = x_realization;
```

Game Traversal

```
function [ui, global_data] = CFR_AS(h, player_i, q, numChoices, ...
    epsilon_1, epsilon_2, model1, model2, t, global_data)
% This algorithm comes from [0] with input from [1] pg. 12
% Monte Carlo variant is from [2]
% h          : current sequence array (call to CFR_AS should be [] to
start)
% player_i   : the player currently playing the game.
% q          : variable needed for MC-CFR sampling (should be 1 to start)
% numChoices : An array saying how many choices player one and player two
%              has at their information sets. (Assumes player two has the
%              same number of actions at all their information sets)
% epsilon_1  : Probability player one plays the current game to a prior
%              model
% epsilon_2  : Probability player two plays the current game to a prior
%              model
% model1     : Model that player one plays to with p=epsilon_1
% model2     : Model that player two plays to with p=epsilon_2
% t          : Current iteration
% global_data: global_data that contains cumulative regrets and
strategies.

% Monte carlo variables;
mc_epsilon = .05;
beta = 10e2;
tao = 100;

%% Find the Information set associated with the current action set
if isempty(h) % This is the chance node that needs to play.
    idx = 1;
    [ui, global_data] = CFR_AS([h idx], player_i, q, numChoices, ...
        epsilon_1, epsilon_2, model1, model2,t, global_data);
    return;
elseif length(h) ==1 % This means it's player ones turn.
    I = 1;            % They only have one information set.
    r = rand();
    if r < epsilon_1
        sig_t = model1{I};
    else
        sig_t = regretMatching(global_data.regret{1},numChoices(1));
    end
else % This is player two's turn.
    try
        I = global_data.InfoSets(h(2));
        % the way the recursion works, this will be reached even if
        % it is a leaf. Make if statement to avoid unneccesary comp.
        if length(h) ~= 3
            r = rand();
            if r < epsilon_2
                sig_t = model2{I};
            else
                sig_t = regretMatching(global_data.regret{2}{h(1)}{I},...
                    numChoices(2));
            end
        end
    catch
        error('ERROR');
    end
end

%% Return the utility
% If player 2 already played, then this is end of the game.
if length(h) == 3
    if h(1) == 1 || player_i == 1 % Smart player knows the risks.
        current_risks = global_data.subRisks_smart;
        current_risks(global_data.ProtInfo(h(2),:,1)) = 0;
        ui = current_risks(h(3));
        %         ui = full(Amat(1+h(2), numChoices(2)*(I-1)+1+h(3)));
    else % Dumb player thinks there are other risks.

    end
    if player_i == 1
        ui = -ui;
    end
    ui = ui/q;
    return;
end
```

```
%% Initialize counterfactual values to zero depending on the player.
(size)
if length(h)==1
    v_sigma_counterfactual = zeros(1,numChoices(1));
else
    v_sigma_counterfactual = zeros(1,numChoices(2));
end

%% Walking through the tree to get counterfactual regrets
% Recursive call to self to figure out the counterfactual values. Note
that
% for nodes that lead to terminal leaves, this is simply the utility of
% that leaf. For leaves beforehand, it is like an expected utility.

% This was all gotten from [2] algorithm.
if length(h) == player_i
    if player_i == 1
        currStrat = global_data.strat{1};
    else
        currStrat = global_data.strat{2}{h(1)}{I};
    end
    r = rand(size(sig_t));
    p  = min(max((beta + tao*currStrat)/(beta+sum(currStrat)),
mc_epsilon),1);
    %     p  = max(max((beta + tao*currStrat)/(beta+sum(currStrat)),
mc_epsilon),1);
    idx = find(r < p);
    for a = 1:length(idx)
        [v_sigma_counterfactual(idx(a)), global_data] = ...
            CFR_AS([h, idx(a)], player_i, q*p(idx(a)), numChoices, ...
            epsilon_1, epsilon_2, model1, model2, t, global_data);
    end
else
    % Update the strategies during the opponents traversal.
    if length(h) == 1
        global_data.strat{1}  =  global_data.strat{1} + sig_t/q;
    else
        global_data.strat{2}{h(1)}{I} = global_data.strat{2}{h(1)}{I}...
            + sig_t/q;
    end
    % Sample one action from opponents current strategy.
    cum_probs = cumsum(sig_t);
    r = rand;
    idx = find(cum_probs>r,1,'first');
    [ui, global_data] = CFR_AS([h idx], player_i,  q, numChoices, ...
        epsilon_1, epsilon_2, model1, model2, t, global_data);
    return;
end

if length(h) == 1
    global_data.regret{1} = global_data.regret{1}  + ...
        v_sigma_counterfactual - sig_t*v_sigma_counterfactual.';
else
    global_data.regret{2}{h(1)}{I} = global_data.regret{2}{h(1)}{I}  +
...
        v_sigma_counterfactual - sig_t*v_sigma_counterfactual.';
end

ui = sig_t*v_sigma_counterfactual.';

end
%% References
% [0] "Regret Minimization in Games with Incomplete Information"
% [1] "An Introduction to Counterfactual Regret Minimization" pg. 12
% [2] "Efficient Monte Carlo Counterfactual Regret Minimization in Games
with Many Actions"
% [3] "Generalized Sampling and Variance in Counterfactual Regret
Minimization"
```

OPF Risk Augmentation

```
mpc_opf = mpc;

% Initialize matrix sizes
nb = size(mpc.bus,1);
nl = size(mpc.branch,1);
ng = size(mpc.gen,1);
nx = 2*nb+2*ng;
N = speye(nx, nx);

% Create quadratic x^T*H*x = x'B*B'*x
[Bbus, Bf, Pbusinj, Pfinj] = makeBdc(mpc);

% This creates the base power reduction
quadratic matrix
B_prime_T = sparse( Risk * [Bf, zeros(nl,nb),
zeros(nl,2*ng)] );

% Find the H matrix (Part of Q)
H_prime = (B_prime_T.'*B_prime_T);

% Create the regularization.
% min (x - x0)^2
% -> min (x - x0)^T * (x - x0)
% -> min (x^T*x - 2x0^T*x + x0^T*x0)
% -> min (x^T*x - 2x0^T*x)
% -> min (x^T * I * x - 2 * x0^T * x)
%
% C = -2 * x0^T
% Q = H' + I
if regularization_flag
    Q_matrix = 2*(H_prime +
R_weight*R_weight*speye(size(H_prime)));

    C = -
2*R_weight*R_weight*[mpc_reg.bus(:,VA)/180*pi;
...
        mpc_reg.bus(:,VM);...
        mpc_reg.gen(:,PG)/mpc_reg.baseMVA; ...
        mpc_reg.gen(:,QG)/mpc_reg.baseMVA];
else
    C = zeros(nx,1);
    Q_matrix = H_prime;
end

%% Weight the generator costs and H by alpha
% min [( 1 - alpha ) * f(x) + ( alpha ) * ( 1/2
x^T * Q * x + C^T * x )]
mpc_opf.gencost(:,5:7) = mpc.gencost(:,5:7)*(1-
probability_OPF_alpha);
Q_matrix = Q_matrix*alpha;
C = C*alpha;

% Initialize new OPF at solution of previous.
mpc_opf.mpopt.opf.init_from_mpc = 1;

% Run the OPF
opf_results =
opf(mpc_opf,sparse([]),sparse([]),sparse([]),mpc
_opf.mpopt,N,sparse([]),Q_matrix,C);
```

CounterFactual Regret Minimization

```
function strategies = getStrategies(T, numCombos,numInfoSets, N, ...
    player_one_model, player_two_model, epsilon_one, epsilon_two, ...
    probability_smart, data)
% This function finds the solution to a game tree given in data through the
% protection and information sets in data.
%
% It contains two functions:
% 1. getStrategies - Sets up the data neccessary and loops through CFR calls.
% 2. CFR_AS - Recursive function meant to traverse the game tree to
%               calculate regrets and strategies through a Monte Carlo
%               sampling approach.
%
% INPUTS:
%
% T               : Number of iterations
% numCombos       : Number of combinations for player one
%                   (should be 3^N if cases aren't eliminated for cost/domination)
% numInfoSets     : Number of information Sets for player two (should be 2^N)
% N               : Number of substations to consider
% player_one_model : A prior model for player one followed with some probability.
%                   Should be one probability vector across all player one's choices.
% player_two_model : A prior model for player two followed with some probability.
%                   Should be a (numInfoSets x 1) cell array of probability
%                   vectors across player two's choices at each of their Information Sets.
% epsilon_one     : probability that player one plays with his prior given
%                   in player_one_model
% epsilon_two     : probability that player two plays with his prior given
%                   in player_two_model
% probability_smart: Meant for a rational attacker with incorrect risk data.
% data            : Struct that contains important information
%    data.substation_risks : (1xN) array of risks associated with substations
%    data.Pmat             : 3D binary matrix who's rows contain system states,
%                            columns are substations, and 3rd layer is the truly
%                            protected and faked layers. Pmat(i,j,1) is
%                            whether substation j is truly protect at
%                            system state i. Pmat(i,j,2) is analogous for
%                            faked. (0=no, 1=yes)
%
%    data.InfoSets        : array wthat corresponds with system states
%                           given in Pmat with which distinguishable state
%                           it belongs to for the attacker.
%
% OUTPUTS:
%
% strategies_one          : Global data with the cumulative regrets and
%                           strategies. The overall 2e-NE strategies can
%                           be gotten by finding the average of
%                           strategies_one.strat
subRisks_smart = data.substation_risks;
InfoSets = data.InfoSets;
ProtInfo = data.Pmat;
store_t = 50;
plot_t = 500;
first_print_flag = true;

%% Figure out upper bounds for CFR
del = max(subRisks_smart);
A1 = numCombos;
A2 = length(subRisks_smart);
I1 = 1;
I2 = numInfoSets;
if probability_smart < 1.0
    I2 = 2*I2;
end
Bound1 = del*sqrt(A1)*I1;
Bound2 = del*sqrt(A1)*I2;
bounds = zeros(T,2);

% Counterfactual Regret Minimization
s1 = zeros(1,numCombos);
r1 = zeros(1,numCombos);

if probability_smart < 1.0
    r2 = {cell(numInfoSets-1,1),cell(numInfoSets-1,1)};
    s2 = {cell(numInfoSets-1,1),cell(numInfoSets-1,1)};
    [r2{1}{1:end,1}, r2{2}{1:end,1}, s2{1}{1:end,1}, s2{2}{1:end,1}] = deal(zeros(1,N));
else
    r2 = {cell(numInfoSets-1,1)};
    s2 = {cell(numInfoSets-1,1)};
    [r2{1}{1:end,1}, s2{1}{1:end,1}] = deal(zeros(1,N));
end
regret = {r1,r2};
strat = {s1,s2};

if isempty(data.global_data)
    global_data = struct();
    global_data.regret = regret;
    global_data.strat = strat;
    global_data.subRisks_smart = subRisks_smart;
    global_data.ProtInfo = ProtInfo;
    global_data.InfoSets = InfoSets;
    global_data.t = 1;
else
    global_data = data.global_data;
end

numChoices = [numCombos, N];
counter = 1;
plots = [];
bounds = [];
figure(1); hold on; pause(0.001);

plot_flag = false;
tic;
for t = global_data.t:global_data.t+T
    %% Actual CFR Loop
    for player = 1:2
        if player == 1
            [~, global_data] = CFR_AS([], player, 1, numChoices, epsilon_one, epsilon_two,...
                player_one_model, player_two_model, t, global_data);
        else
            [~, global_data] = CFR_AS([], player, 1, numChoices, epsilon_one, epsilon_two,...
                player_one_model, player_two_model, t, global_data);
        end
    end
    global_data.t = global_data.t + 1;
    % Store the results every now and then to plot
    if mod(t,store_t)==0
        plots(counter,1) = t;
        plots(counter,2) = max(max(global_data.regret{1}),0)/t;
        temp_regret = 0;
        for  tr = 1:length(global_data.regret{2})
            temp_regret = temp_regret + ...
                sum(max(cellfun(@max,global_data.regret{2}{tr}),0));
        end
        plots(counter,3) = temp_regret/t;
        bounds(counter,1) = Bound1/sqrt(t);
        bounds(counter,2) = Bound2/sqrt(t);
        counter = counter + 1;
    end
    % Plotting every now and then.
    if mod(t,plot_t)==0
        t2 = toc;
        if ~plot_flag
            handle1 = plot(plots(1:counter-1,1),plots(1:counter-1,2));
            handle2 = plot(plots(1:counter-1,1),plots(1:counter-1,3));
            plot_flag = true;
        else
            set(handle1,'Xdata',plots(1:counter-1,1));
            set(handle2,'Xdata',plots(1:counter-1,1));
            set(handle1,'Ydata',plots(1:counter-1,2));
            set(handle2,'Ydata',plots(1:counter-1,3));
        end

        set(gca,'yscale','log');
        set(gca,'xscale','log');
        %       set(gca,'xscale','log');
        legend('P1 Regret', 'P2 Regret');%,'P1 Bound','P2 Bound');
        grid on;
        drawnow();
        % Find the time left.
        perc_left = (1-t/T)*100;
        avg_time_per_percent = t2/(t/T*100);
        hours = floor(perc_left*avg_time_per_percent/3600);
        minutes = floor((perc_left*avg_time_per_percent - hours*3600)/60);
        seconds = perc_left*avg_time_per_percent - hours*3600 - minutes*60;
        msg = sprintf('Iteration: %d, Time Left: %d:%d:%3.2f\n',...
            t,hours,minutes,seconds);
        reverseStr = repmat(sprintf('\b'),1,length(msg));
        if first_print_flag == false
            fprintf([reverseStr, msg]);
        else
            fprintf(msg);
            fprintf('\n');
            first_print_flag = false;
        end
    end
end
toc
% Find the average strategies that is guaranteed to converge to 2e-NE
global_data.strat{1} = global_data.strat{1}/sum(global_data.strat{1});

for tr = 1:length(global_data.strat{2})
    for ii = 1:length(global_data.strat{2}{tr})
        global_data.strat{2}{tr}{ii} = ...
            global_data.strat{2}{tr}{ii}/sum(global_data.strat{2}{tr}{ii});
    end
end

strategies = global_data;
end
```