AN ABSTRACT OF THE THESIS OF

Austin R. Lassetter for the degree of Master of Science in Electrical and Computer Engineering presented on June 7, 2021.

Title: Enhancing Resolution and Scalability of Power Grid Reliability Algorithms

Abstract approved: _____

Eduardo Cotilla-Sanchez

This Thesis aims to determine whether we can improve the accuracy, resolution, and speed of calculations for common power system problems using simple computational models that scale well to machine learning and high performance computing solutions. The second chapter of this Thesis implements more precise aging and degradation models for power grid equipment for a more precise Contingency Analysis. Contingency Analysis, a widely used reliability metric in the grid, determines whether a single (or multiple) transmission or generator outage will place the grid in an unstable state. After the aging models are defined, aging related failure rates of the models are integrated into the grid by constraining power limits. Contingency Analysis experiments are then performed, determining whether increasing the age of the grid will increase the number of contingencies on the grid. This effectively defines Contingency Analysis simulations that are more robust with respect to deterioration of equipment, traditionally ignored in this type of

simulation. The third chapter applies Critical Slowing Down, a statistical method that can be used as a feature engineering technique, the process of extracting key properties of data, to Phasor Measurement Units (PMU). In this case, Critical Slowing Down has been shown to be indicative of critical transitions in power systems data. Critical Slowing Down analysis is applied to the data to enhance the training time, accuracy, and time localization of Artificial Neural Network classification of events on the grid, by allowing easier localization of critical transitions. The fourth chapter compares GPU architectures and CPU architectures for the implementation of widely used machine learning algorithms for PMU data. GPU architecture leverages mass parallelization, which is useful for many complex algorithms. However, CPU architecture is much faster for serial processes, which many of the algorithms are also based upon. Many of these algorithms contain some mix of serialization and parallelization, making it generally unclear on whether CPU architecture or GPU architecture is more applicable for the algorithm at hand. This chapter compares the computational performance of GPU architectures and CPU architectures of commonly used machine learning algorithms for PMU data. The dataset sizes are varied for each of the algorithms determining whether CPU architectures or GPU architectures scale better for large datasets.

Material from this thesis has been submitted or accepted for publication in the following form:

Lassetter, A. and Cotilla-Sanchez, E. and Kim, J., Using Critical Slowing Down Features To Enhance Performance of Artificial Neural Networks for Time-Domain Power System Data, International Conference on Smart Grid Energy Engineering, 2021

AND

Lassetter, A. and Cotilla-Sanchez, E., Exponential Modeling of Equipment Degradation in the Grid for More Reliable Contingency Analysis, International Conference on Smart Energy Systems and Technologies, 2021

AND

Lassetter, A. and Cotilla-Sanchez, E., Accelerating Phasor Measurement Units (PMUs) Data Analytics, MIT A+B" Applied Energy Symposium, 2021

Enhancing Resolution and Scalability of Power Grid Reliability Algorithms

by

Austin R. Lassetter

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 7, 2021
Commencement June 2022

Master of Science thesis of Austin R. Lassetter presented on June 7, 2021.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Head of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

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

Austin R. Lassetter, Author

ACKNOWLEDGEMENTS

I would like to thank my adviser, Eduardo Cotilla-Sanchez for his continued guidance throughout the projects and for helping me advance toward my goals in research and my professional career. I would also like to express my thanks to him and many other professors at Oregon State who have helped me gain and utilize skills and understand concepts that would prove to be critical in my research. I express gratitude toward the financial support that the US Department of Energy (DOE) provided for parts of these projects, along with the support of the university. I would like to express gratitude to my colleagues and friends who I have worked with in classes and projects. Finally, I would like to thank my parents, brother, and sister, who have all been great role models and have provided me with a healthy and competitive learning environment.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: Introduction

The electric grid is one of the most important infrastructure networks in the United States and will be of even greater importance in the near future. Communities are becoming more reliant on electricity making the electric grid critical to protect. Residences and commercial buildings are now using electricity as the main energy resource over gas and oil. Transportation is becoming more reliant on electricity, with vehicles and even railways transitioning toward electricity rather than fossil fuels. Forecasts show that electric energy usage will increase if electrification of industries is highly incorporated in the near future, even with more tightly controlled energy efficiency measures [38]. People are now more connected than ever with smartphones and other personal computing devices, allowing for long distance communication. Many tedious tasks are being automated and many tools that are being used by industries are directly connected to electrical devices. Without reliable electricity distribution, our healthcare system, economy, and daily lives would become much different than it is today. For example, particular events like a global pandemic would highlight our reliance on reliable electricity distribution. Widespread power outages could block communication between governments and their people, which could exacerbate the situation. Additionally, Healthcare facilities and transportation may not be able to supply those in need with supplies and services in time. Without reliable electricity distribution, many situations

in modern life would be difficult to comprehend. This reliance on electricity will only increase, as many upcoming industries and software tools are heavily driven by it. For example, Figure 1.1 shows that training a huge AI model with neural architecture search for long periods of time can take the five times the amount of energy (measured in CO_2 emissions) as a car in its entire lifetime [47].

The grid is also a physical system of thousands of electrical components that are subject to weathering and failure. Similar to any other system, the grid's individual components will eventually fail, placing additional stress on the electric grid and possibly causing it to become unstable. In extreme cases a transmission line or some other equipment failure will cause additional stress on other equipment, leading to a chain of component failures across the grid. This is known as a cascading failure, and can lead to outcomes such as blackouts. For reference, one of the larger blackouts, the Northeast US blackout, cost over $5 billion to the US economy [2].

The electric grid is also the largest interconnected system in the United States, connecting residences, industries, and other facilities directly or indirectly to distant electricity generation plants. This grid is composed of thousands of transmission lines, transformers, generators, and other devices that allow the electricity to be produced, to flow across wide areas, and to be transformed into usable power levels based on the consumer's needs. However, this system is much more complex than the network alone. The grid will become unstable if there is more generated electricity in the grid than electricity demanded or vice versa. Additionally, the grid is not owned by a single entity. Rather, the grid is maintained region-

Consumption	CO_2e (lbs)
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Figure 1.1: Benchmarking of AI CO_2 emissions compared to lifestyle CO_2 emissions [47]

ally be various utilities and thus is deregularized. Order 888 made transmission services open access, which effectively split transmission services and generation services [12]. Additionally, Order 2000 formed Regional Transmission Organizations (RTOs), such that all utilities within an RTOs region must act accordingly to their RTO [13]. As a result, individual utilities want to ensure that their generation or transmission is online to maximize profits, but had to act accordingly to their RTOs. Furthermore, the RTOs have to communicate between each other as well as the grid is still interconnected between RTOs. These issues create an even more complex network, especially at the communication level.

Due to the complexity of the network, Extensive precautions and research have been and are being done to reduce the likelihood that the electric grid becomes unstable, leading to a blackout. Reliability metrics, such as contingency analysis (CA), are used in facilitation of the grid, alongside verifying the current stability

of the grid. These reliability metrics are critical for scheduling planned outages (such as scheduling a transmission line outage for construction) and to ensure that an unplanned outage won't lead to unstable conditions on the grid. Additionally, the rise of Phasor Measurement Units (PMUs) provide real-time data of the current state of the grid. This data consists of voltage, current, frequency, and rate of change of frequency. The PMU data can be leveraged using various machine learning or AI algorithms, which help detect outages and other grid instabilities.

The overall hypothesis of this work is: Can we improve the accuracy, resolution, and speed of calculations for common power system problems using simple computational models that scale well to high performance computing and machine learning solutions?

This thesis expands upon CA and utilize PMU data to reinforce the grid from instabilities. In chapter 2, we develop aging models for equipment and adjust the grid state by reducing power limits according to the progressively aged equipment. This adjusted grid state is then input into an AC Optimal Powerflow (AC-OPF) CA simulation, determining whether age is a factor in the number of contingencies in the CA and giving us a more robust CA simulation.

In chapter 3, we utilize PMU data using Critical Slowing Down (CSD), a statistical metric that has been shown to be indicative of Power System instabilities, and artificial neural networks (ANNs) to detect anomalies within the grid.

In chapter 4, we explore GPU computing of common algorithms used in power systems and compare them to their CPU alternatives. Algorithm efficiency is critical in power system data as data is sampled at high frequencies. Algorithms

that lag behind this frequency may cause the algorithm to detect instabilities too slowly to allow timely maintenance. This chapter will serve as a benchmark, helping operators determine the best hardware setup and algorithm implementations for their controls by comparing and contrasting CPU implementation and GPU implementations of algorithms that are commonly used in power system data.

Chapter 2: Grid Reliability: Equipment Degradation Modeling

As more communities become increasingly dependent on electrical power demand, reliable distribution of power has become vital to our society. The margin for operational errors in the energy delivery infrastructure is very small. Over a decade ago, the 2003, the Northeast US blackout cost over $5 billion to the US economy [2]. A blackout of similar magnitude today would be a catastrophic event to our economy. Smart Grid technologies play an important role in preventing blackouts by incorporating improvements in communications, cyber attack resilience, and implementing energy storage facilities [4, 6, 45]. However, these technologies cannot completely solve reliability issues of the grid. The components of the grid have finite lifespans and internal or external forces can damage the equipment. Extensive research has been done to evaluate the effects of aging, weather, and correlated outages on power grid equipment [8, 11]. Some of these effects may deteriorate components to the point that they are not stable at their maximum or minimum ratings, posing a threat to the stability of to many commonly used reliability metrics, such as Contingency Analysis (CA).

CA is a widely accepted system study that facilitates managing the grid. CA determines how likely a tripped transmission line could lead to cascading failures in the system. Currently, the most common types of contingency violations are either voltage violations or MVA limit violations [35]. Aged or deteriorated electri-

cal components may not be able to tolerate their absolute maximum load ratings, which impacts the validity of a contingency violation. To resolve this, previous research has created probability models for thermal aging and other stress variables to screen buses with high probability for failure, creating better alerts for contingency failures [10]. This chapter expands upon these aging models by introducing models for multiple types of equipment that include individual components of each piece of equipment. The models are then integrated into CA by adapting generator and branch power limits in the simulation.

2.1 Background and Motivation

CA is widely applied to manage power load within the grid and provides insight on whether a branch or generator outage, or another contingency, puts the grid at an unstable state. An $N-1$ contingency represents a generator or branch outage that leads to an unsolvable power flow solution. It is paramount to avoid $N-1$ contingencies propagating into a cascading failure. For this reason, to ensure $N-1$ security, more extensive analyses have also been developed, helping to get a better perspective on the state of the grid simulating additional failures, for example $N-1-1$ contingencies, and $N-k$ contingencies. Due to the computational complexities of these problems, previous research has investigated ways to improve power flow convergence, speed of power flows solutions, and various CA improvement techniques [19, 31, 50].

Although improving the speed of CA is important to allow higher order CA,

these methods fail to account for other factors that may be placed on components, namely component deterioration due to weathering and aging, and disproportionately contribute to risk. As alluded to in the introduction, components can be placed in poor condition due to weathering, aging, or past correlated outages. This chapter explores models for transformer, generator, and transmission line aging and implements them into CA. The models for these equipment are based on research toward aging related failure causes of the equipment and their components, shown in the subsections below.

2.1.1 Transformers

Transformers are critical components that link the distribution network to the transmission network on the grid, so modeling the aging of transformers can be critical in predicting an outage for sub-networks connected in between those transformers.

2.1.1.1 Bushing Insulation

The remnant life of the transformer has a strong exponential relationship with furanic compound presence in transformer oils. Furanic compound presence in transformer oils is caused by insulating paper degradation and is accelerated from faults [3]. This reference also notes that applying voltages above the transformers' rated limits and temperatures also contribute to transformer bushing deterioration.

2.1.2 Generators

Previous research has shown that when generator stator insulation becomes weak, the generator becomes prone to mechanical and electrical failures. We examine and integrate three models of High Voltage generator insulation failure:

2.1.2.1 Electrical stress

Electrical stress has a direct correlation with generator insulation failure. A common relationship between insulation life of dielectrics and electrical stress is given below:

$$L = K_E \cdot E^{-C_E} \tag{2.1}$$

where L is the insulation life, E is a given electrical stress, and K_E and C_E are constants [46].

2.1.2.2 Mechanical stress

Mechanical stressors, such as vibrations, create mechanical defects in the insulation. The pattern for mechanical stress also follows a power law:

$$L = K_S \cdot S^{-C_S} \tag{2.2}$$

where L is the insulation life, S is a given mechanical stress, and K_S and C_S are constants [46].

2.1.2.3 Thermal stress

Electrical and mechanical stressors are amplified by thermal stress, which depends on the thermal expansion coefficient and the temperature of the insulation. Research has found a simplified Arrhenius model that that combines electrical and thermal stress:

$$L(T, E) = K_E(T) \cdot E^{(-C_E(T))} \cdot e^{(\frac{B}{T})} \tag{2.3}$$

where $L(T, E)$ is the insulation life dependent on temperature (T) and electrical stress (E), $K_E(T)$ is a scalar related to temperature, $C_E(T)$ depends on temperature, and B is a constant related to the activation energy of the aging process.

On the other hand, the amount of mechanical stress that the generator stator insulation can withstand is linearly related to the temperature [44]. The slope of this linear relationship is correlated to the insulators' thermal expansion coefficient, α. Adding these relationships into equation two gives us the following combines thermal-mechanical equation:

$$L(T, S, \alpha) = K_S(T, \alpha) \cdot S^{-C_S(T,\alpha)} \tag{2.4}$$

Where $L(T, S, \alpha)$ is the insulation life dependent on Temperature (T) and the thermal expansion coefficient (α), along with mechanical stress (S). $K_S(T, \alpha)$ is a scalar related to temperature and the thermal expansion coefficient, and $C_S(T, \alpha)$ is a function of T and α determining the slope of exponential deterioration.

2.1.3 Transmission Lines

Transmission line conductors lose strength due to annealing over time and the insulators can also degrade over time, exposing the transmission line to more adverse external factors [7]. Both factors affect the transmission line aging model.

2.1.3.1 Conductor loss of tensile strength

Conductors in transmission lines are prone to aging from high voltages and thermal loads due to high power transfer. At high temperatures, the strands in the transmission line's conductor will elongate and lose strength [7]. These losses in strength slowly decrease the transmission line's maximum load over time, possibly leading to failure under high load conditions. It has been observed that the emissivity of the conductor grows over time and the percent loss of tensile strength increases grows over time polynomially with a polynomial degree less than one [7]. It has also been observed that a higher temperature increases the rate at which the tensile loses strength [7]. Existing research has explored a model that relates the loss of tensile strength to a safety factor, SF, which indicates whether or not the loss of tensile strength places the transmission line in critical condition [24]:

$$SF = \frac{RTS \cdot (1 - \frac{LOS}{100})}{MWT} \tag{2.5}$$

Where RTS is the transmission lines rated tensile strength, LOS is the loss of tensile strength, and MWT is maximum working tension under worst case load

conditions.

2.1.3.2 Insulation loss

Insulation loss of transmission lines make the electrical components of a transmission line prone to failure due to external factors. To determine the deterioration of insulation in transmission lines, the tangent of the loss angle has been widely used [5]. The loss angle shifted from 90° indicates a loss of insulator resistance, which influences dielectric losses [5]. A field study was explored determining the rate of loss angle shift over a 30 year period for 15 transmission lines. The results were consistent and nearly identical for each transmission line, each exhibiting and exponential increase of loss angle over the 30 year time period. An important finding from this study is that the applied voltage has minimal effect on the loss angle increase over time, indicating that the insulator predominantly ages from external factors [5].

2.2 Methodology

Using the research that describes individual component aging, as discussed in Section II, we propose a comprehensive model in this section, and a framework to integrate it into CA. This model then considers the multiple reasons for resulting failures due to overall aging risk. The models are expressed as failure rate functions that are dependent on time, using years as the unit, and the equipment specific

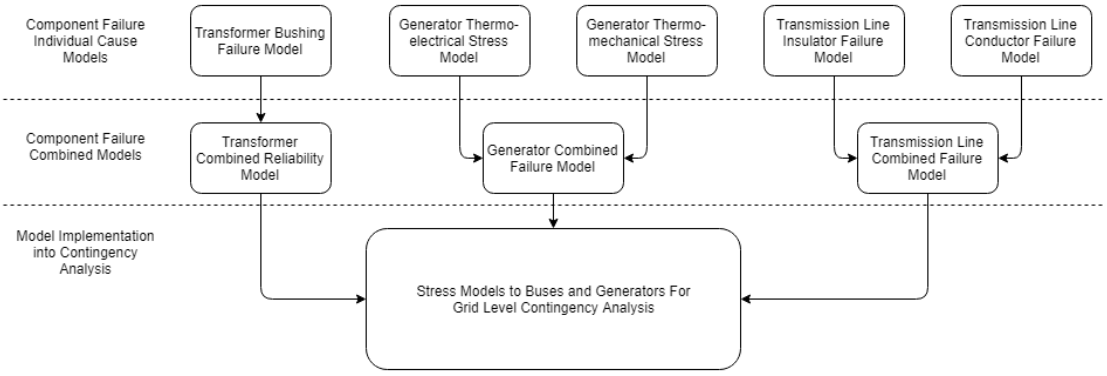

Figure 2.1: Top-Down model approach

parameters. Since is not feasible to model every combination of equipment failures, the failure rate models in turn will reduce the power limits of the respective generators and branches, allowing new contingencies to be revealed, if any.

The aging models for the components of equipment are modeled as Weibull distributions defined by equipment-specific parameters. The shape parameter β_X is a constant for each Weibull distribution and the scale parameter η_X is a function of the specific parameters of component type X to scale the Weibull distribution. The Weibull distribution is described as follows, with X representing the type of component:

$$f(t) = \frac{\beta_X}{\eta_X} \cdot (\frac{t}{\eta_X})^{\beta_X - 1} \cdot e^{-(\frac{t}{\eta_X})^{\beta_X}} \tag{2.6}$$

The Weibull distribution's failure rate function for the type of component is then given as:

$$\lambda_X(t) = \frac{\beta_X}{\eta_X} \cdot (\frac{t}{\eta_X})^{\beta_X - 1} \tag{2.7}$$

To define Weibull distributions for each type of equipment, η_X is defined as a

function of parameters of component X, providing a failure rate function for component X when given a set of parameters for the component and the Weibull shape parameter β_X.

We implement the CA using the open source library MATPOWER to evaluates each of the model's performance [63]. Figure 2.1 depicts the overall methodology in a flowchart. The models are simulated across multiple aging components, where the age and parameters involved in the aging models have Gaussian distributions. The contingencies of these aged models are then compared with a CA with no aging for baseline benchmarking.

2.2.1 Transformers

Based on the furanic compound analysis, an exponential model, and thus a Weibull distribution, best models bushing deterioration. Temperature rise ratings, T_R, and the transformer's insulation temperature rating, T_I, are important parameters in determining the rate of exponential deterioration.

The following equation represents the scale parameter η_X:

$$\eta_X = \phi_{temp} \cdot \xi_X \cdot e^{\frac{1-\delta_X}{K_X}} \tag{2.8}$$

Where ξ_X is the baseline lifespan coefficient of transformers, δ_X is an index for the likelihood of faults, K_X is a constant that controls the rate of growth, and ϕ_{temp} is the ratio of temperature rise and insulation ratings ($\frac{T_I}{T_R}$).

2.2.2 Generators

From Section II, B, Equation 2.3 introduces a mixed thermal-electrical stress model on the insulation life of the stator, while Equation 2.4 introduces a mixed thermal-mechanical stress model on the insulation life. These equations signify that higher electrical and mechanical stresses shortens the lifespan of the insulation exponentially. The rate of the decay is directly related to the temperature, T, in both models. The interactions between these stress models are assumed to be small, so the models are assumed to be independent.

2.2.2.1 thermal-mechanical stress model

The Weibull distribution for the thermal-mechanical model includes the insulation's thermal expansion coefficient, α, and the average mechanical stress that the insulation endures, $\bar{S}$, with respect to the rated stress. Additionally, σ_S is introduced to account for high stress situations. Temperature of the conductor is correlated to the load of the conductor (L_C), which is expressed as the average load divided the rated load. σ_{LC} represents the likelihood of the conductor being in an overloaded state, which accelerates conductor deterioration.

The following equation represents the scale parameter η_{tm} of the thermal-mechanical stress model's Weibull distribution:

$$\eta_{tm} = \frac{K_{tm} \cdot e^{(-(2\cdot(L_C+\sigma_{LC})))} \cdot e^{(-(2\cdot(\bar{S}+\sigma_S)))}}{\alpha} \tag{2.9}$$

Where K_{tm} is a constant.

2.2.2.2 thermal-electrical stress model

The Weibull distribution for the thermal-electrical model includes the insulation's temperature (T), the average electrical stress that the insulation endures ($\bar{E}$), and the constant related to the activation energy (B). Additionally, to account for the insulation stress variability over time, σ_E is introduced for faster deterioration of heavily over-stressed insulation. Temperature of the generator is correlated to the load of the generator (L_I), expressed as the average load divided the rated load. σ_{LI} represents the likelihood of the insulator being in an overloaded state, which accelerates insulation deterioration.

The following equation represents the scale parameter of the thermal-electrical stress model's Weibull distribution, η_{te}:

$$\eta_{te} = K_{te} \cdot e^{(-(2*(L_I+\sigma_{LI})))} \cdot e^{(-(2*(\bar{E}+\sigma_E)))} \cdot e^{\frac{B}{L_I}} \tag{2.10}$$

2.2.2.3 Combined model

The thermal-electrical stress and thermal-mechanical stress models are combined by multiplying the rates of the insulation and stator properly working. This gives the probability that the generator has no failures, shown in the equation below:

$$P_{gen}(t) = \max(1 - \lambda_{tm}(t), 0) \cdot \max(1 - \lambda_{te}(t), 0) \tag{2.11}$$

This equation provides the likelihood that the generator's stator and insulation both function properly. A max function is included to account for the failure rate rising above one. In those cases, the insulation or stator are assumed to be non functional, rendering the generator obsolete. The failure rate of the generator can be expressed as follows:

$$\lambda_{gen}(t) = 1 - P_{gen}(t) \tag{2.12}$$

2.2.3 Transmission Lines

The explored models of conductor tensile strength loss and insulator loss provide great details of how transmission lines age over time. Models for both the conductor tensile strength loss and insulator loss are created using research gathered in section III, and then the models are combined for a more robust model of transmission line aging.

2.2.3.1 conductor tensile strength loss

From Section II, C, the safety factor is introduced giving a predicative indication of whether a transmission line is in critical condition due to conductor tensile strength loss, repeated below:

$$SF = \frac{RTS \cdot (1 - \frac{LOS}{100})}{MWT} \tag{2.13}$$

The safety feature and reliability of the conductor tensile strength is directly related to the Rated Tensile Strength (RTS) and the Maximum Working Tension (MWT). Additionally, a higher temperatures increases the rate at which the conductor tensile loses strength. The variable $\bar{L_S}$ is introduced to represent the average load with respect to the transmission lines rated load. Additionally, σ_{LS} is introduced to account for variance of the line load. $\bar{L_S}$ and σ_{LS} is used to account for the temperature of the transmission line.

A Weibull distribution will represent the exponential conductor tensile strength loss with the scale factor as follows:

$$\eta_{txc}(t) = \frac{K_{txc} \cdot RTS \cdot e^{(-2 \cdot (\bar{L_S} + \sigma_{LS}))}}{MWT} \tag{2.14}$$

Where K_{txc} is a constant.

2.2.3.2 transmission line insulation

Previous research has found that applied voltage to the transmission line does not have an impact on loss angle growth over time, suggesting that the conductor insulation ages naturally over time. A simple Weibull distribution is an accurate way of modeling the failure rate due to insulation deterioration. In this Weibull distribution the scale factor, η_{txi}, is simply a constant representing the conductor insulation's natural aging scale.

2.2.3.3 combined model

The insulation and the conductor tensile strength loss model are combined by multiplying the rates of the insulation and tensile properly working. This gives the probability that the transmission line has no failures, as given in the equation below.

$$P_{TX}(t) = \max(1 - \lambda_{txc}(t), 0) \cdot \max(1 - \lambda_{txi}(t), 0) \tag{2.15}$$

This equation provides the probability that the transmission line functions properly given the current tensile strength loss and the insulation loss. A max function accounts for the failure rate rising above one. In those cases, either the tensile cannot withstand stress or the insulation has failed, placing the transmission line in critical condition. The failure rate of the transmission line can then be expressed as follows:

$$\lambda_{TX}(t) = 1 - P_{TX}(t) \tag{2.16}$$

These models are integrated into the power flow simulations assuming that the likelihood of equipment failure increases the constraints on power for the buses, generators, and branches that the component is attached to. Using the assumption that an $X\%$ failure rate represents an $(1 - X)\%$ in power constraint, the following implementations are used for the generator, transformer, and transmission lines.

2.2.4 Generator Implementation

The generator implementation lowers the maximum active per unit power output with respect to the minimum power output.

$$P_{GMax} = P_{GMax} - (\lambda_{gen}(t) \cdot (P_{GMax} - P_{GMin})) \tag{2.17}$$

Where P_{GMax} and P_{GMin} are the maximum and minimum active power output measured in p.u.

2.2.5 Transformer Implementation

The transformer implementation restricts the MVA limits of the branches connected to the transformer's bus. Since multiple buses connect to the same branch, the minimum MVA limits of each branch are kept to account for the weakest link.

$$MVA(\bullet) = MVA(\bullet) \cdot (1 - \lambda_X(t)) \tag{2.18}$$

2.2.6 Transmission Line Implementation

The transmission line implementation is a simple implementation, where a transmission lines A, B, and C MVA ratings (long term, short term, and emergency, respectively), denoted by $MVA(\bullet)$ are attenuated by the failure probability. Since these ratings also are impacted by the transformer implementation, the minimum

MVA limits of each branch are retained.

$$MVA(\bullet) = MVA(\bullet) \cdot (1 - \lambda_{TX}(t)) \tag{2.19}$$

2.3 Verification

To create a more robust model for equipment, Gaussian distributions for the equipment specific parameters are used to incorporate variability within the network.

We verify the model sensitivity analsysis by simulating N-1 contingencies with MATPOWER 7.0 ac Optimal Power Flow (ac-OPF) method under different controlled aging levels. Since the models are exponential, honing into specific time frames of equipment damage is important as the number of contingencies and failures hypothetically begin to rise quickly as it approaches the anticipated age of failure. To account for these ages, Gaussian distributions of ages will be applied to each type of component separately, centered around 7 different aging levels. These aging levels progress from level 1, where the equipment has starting age, to level 7, where the equipment is past their anticipated failure age. Additionally, three separate experiments are performed for the individual models of each type of equipment. These experiments will only apply aging to their respective type of equipment and will progressively increase the equipment age until ac-OPF failure.

Before the aging simulation, an ac-OPF without any branch outage is simulated to verify that the Gaussian component aging did not age critical components to the point where there is no feasible power flow solution. If this is the case, the

component ages are re-randomized drawn from the Gaussian aging distribution until either reaching 100 failed power flows or a power flow solution occurs.

2.4 Results

As indicated above, multiple cases of CA using Matpower 7.0 AC-OPF were ran under the 7 aging levels to evaluate the equipment models. The test cases under study are widely used benchmarking networks of a variety of sizes, namely case30, case30Q, case39, case_RTS_GMLC, case1354pegase, case2383wp, case2736sp, case-2737sop, and case2746wp [28] [21].

Figure 2.2 shows that simulating equipment aging on the grid creates more contingencies than the non-aged baseline. All cases show that the number of contingencies grow slowly at first and then after a critical threshold they exponentially grow until the the aging causes the ac-OPF to not converge. Figures 2.3, 2.4, and 2.5 show the individual equipment models in isolation. Similar to the combined model, these figures show that contingencies generally increase as the age is increased for all models. These fail at different ages for different cases, due to differences of the grid state. The transformers and transmission line contingency curves trace a much more clear trend than the ones from the generators, which is due to the randomness of initializing the component parameters.

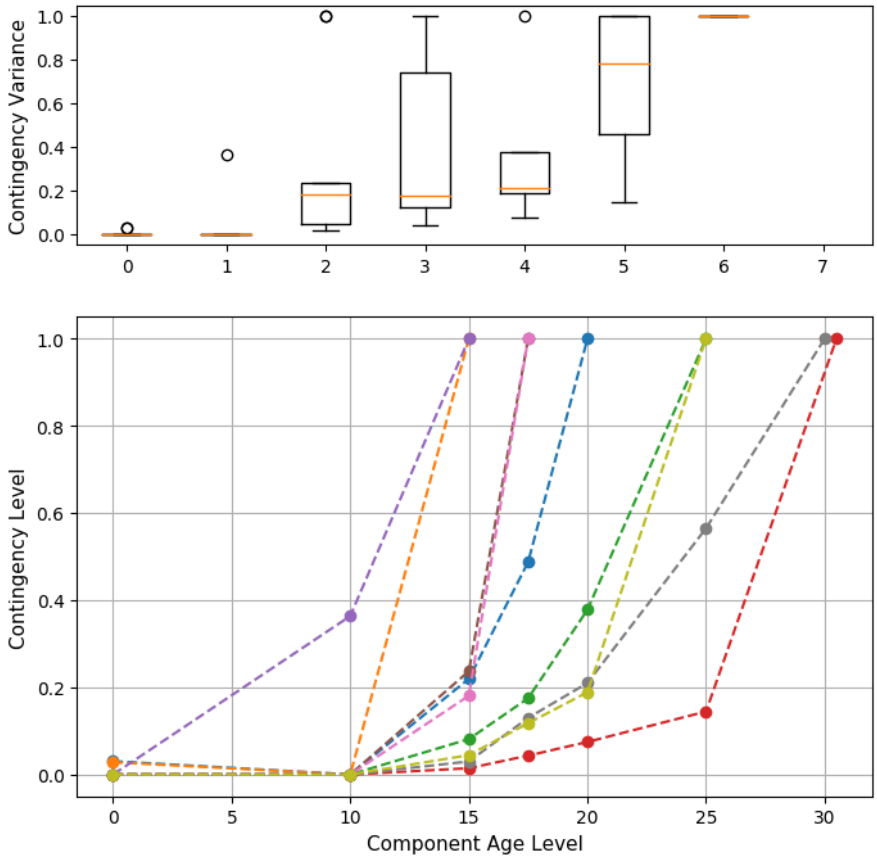

Figure 2.2: Contingency Table With Aging Levels Applied to All Components

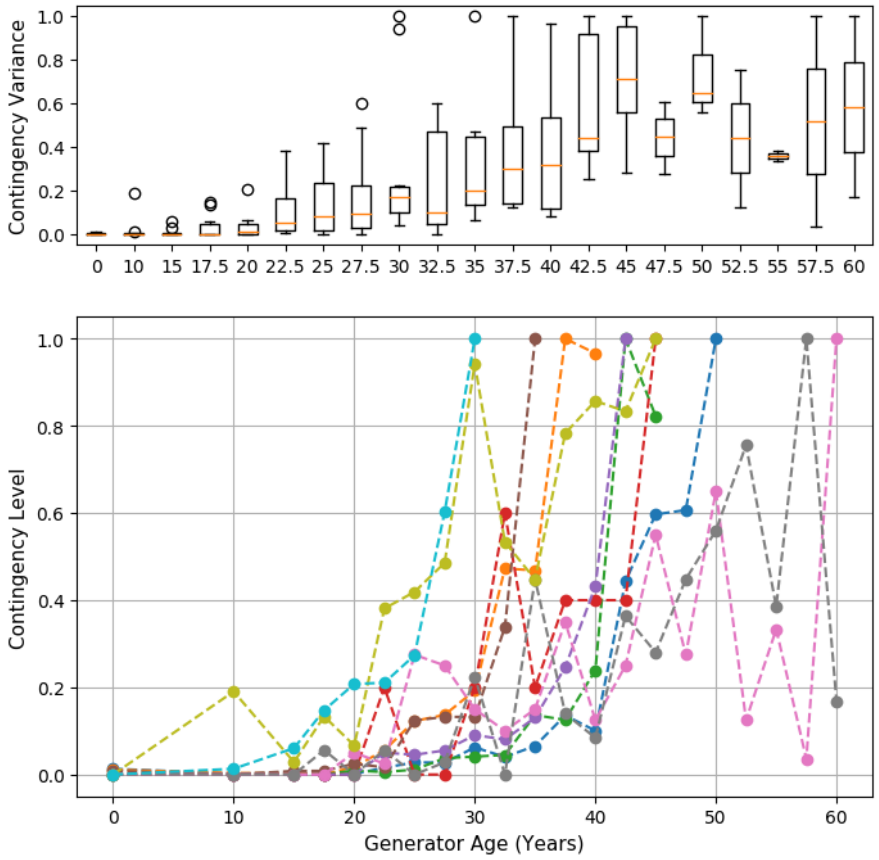

Figure 2.3: Aging Level vs. Contingency - Generator

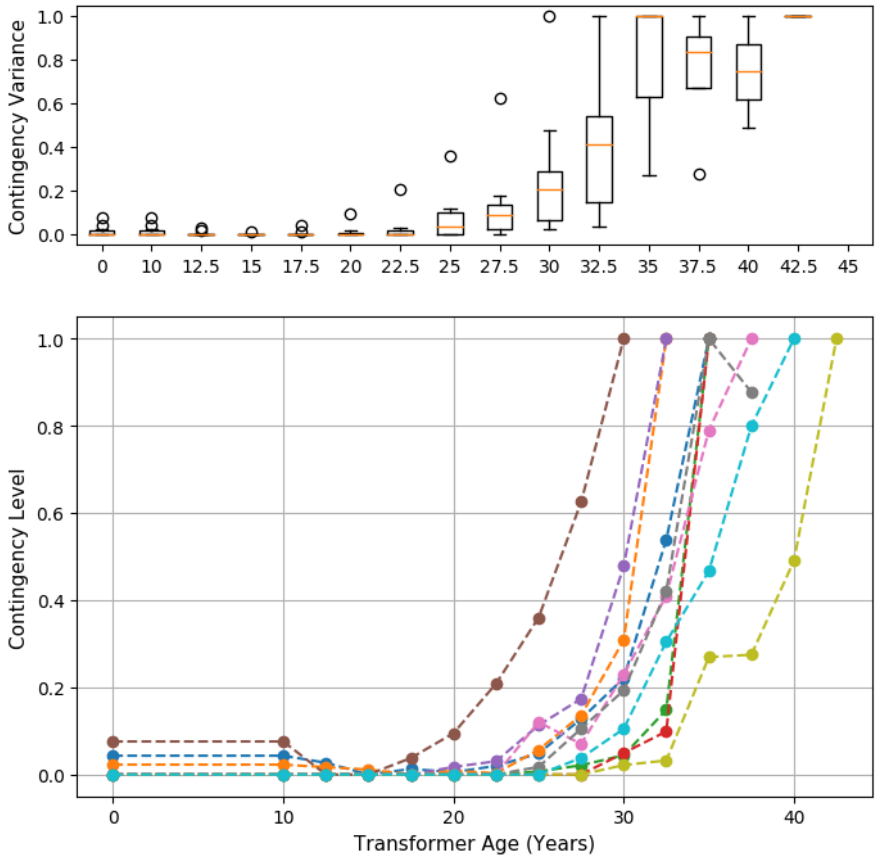

Figure 2.4: Aging Level vs. Contingency - Transformer

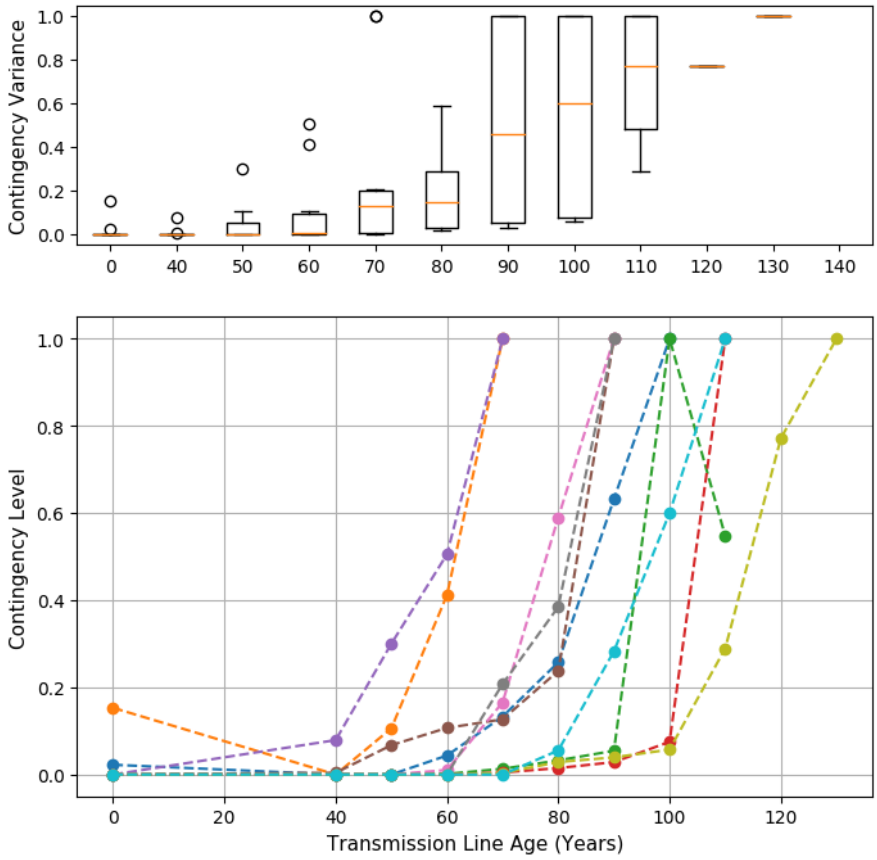

Figure 2.5: Aging Level vs. Contingency - Transmission Line

2.5 Conclusion and Next Steps

Electrical equipment aging in the grid is one of the many reasons for bus, generator, and branch failures across an electrical grid. This analysis proposes to integrate models for electrical equipment aging for transmission lines, generators, and transformers. The models are transformed from failure rate functions to additional power limitation on branches and generators within the power flow analysis. Example N-1 contingencies were then simulated across multiple cases and different aging conditions to benchmark the integrated model. All of the cases show an exponential growth of failures around the point where the equipment nears its nominal full life expectancy. All cases also show excessively aged components result in violations on the ac-OPF, indicating that every outage is a violation. The results of these experiments suggest that standard CA without combining multiple individual aging effects may underestimate the number of contingencies in the grid. Using at a minimum a basic form of aging or degradation model integrated with power flow models will help reduce the uncertainty of the number of contingencies on the grid for multiple other applications.

Some limitations of the models include a weak pre-contingency screening of the grid. Future research should find ways to identify the most critical components on the grid with more up-to-date condition data in order to find the more salient scenarios of aging related contingencies.

Chapter 3: Anomalies Within the Grid: Critical Slowing Down

With the increasing demand for power and electricity, reliability on the power grid is at an all time high. Researchers have been heavily involved in developing preventative methods to both secure and maintain the grid's status, such as advances in computational performance and sensitivity in contingency analysis [52, 59]. The drawbacks to contingency analysis is that it does not always account (in a computationally efficient way) for the critical line or transformer outages on the grid, resulting in limited results that could cause more issues.

The rise of Phasor Measurement Units (PMUs) help us maintain and secure the grid status by providing real-time data about the state of the grid. This data has allowed researchers to heavily explore statistical patterns in data that results from different grid states. Heavy research efforts have been put into using machine learning and statistical processing methods to leverage this PMU data and identify fault locations or determine events that cause islanding [29, 40, 56]. Research has also explored event detection on the grid for line outages, unusual frequency patterns, and oscillatory events [16, 42].

This chapter aims to expand on event detection by applying critical slowing down (CSD), a statistical transformation method, to the input data [15]. CSD has been shown to be indicative of events on power systems, while also being quick to compute [22]. Additionally, it has been used with machine learning to

predict voltage collapse on synthetic systems with renewable energy generation and has been used as an indication of stability loss in a power system [33] [41]. As an indicator of critical transitions and instability in a power system, applying CSD to PMU data could provide improvements to event classification accuracy as frequency events, oscillation events, and outages create instabilities and critical transitions in the network. Additionally, in the case of artificial neural networks (ANNs), CSD may improve training time of the model as features of events are more distinct after applying CSD to the input data. To evaluate the impact of CSD on the data, two similarly structured ANNs are used to train and evaluate the accuracy of event detection using CSD and without CSD, alongside the training time of each model.

3.1 Background and Motivation

Event detection on the power system has been studied extensively by researchers in order to develop models or algorithms that are able to correctly classify events. The ability to quickly identify events on the grid is critical to determining whether follow-up action is needed after the event to ensure that the grid is stable. Operator or automated failure to address follow-up actions could result in the grid being in an unstable state, possibly leading to cascading failures and widespread blackouts. For example, in 2003, a blackout in the Northeast cost over $5 billion to the U.S. and Canada economy, and left millions without power. Considering that was almost two decades ago and considering that the economy and peoples' lives

revolve more heavily around technology and power, a similar blackout could be magnitudes more catastrophic today than it was in 2003.

Artificial Neural Networks have become increasingly researched for PMU data applications. For example, Spatial Pyramid Pooling was applied to a Convolutional Neural Network (CNN) and achieved high accuracy [54], especially if the data is of high quality (few missing datapoints).

Our research contributes to improving methods to classify power related events using PMU data by using feature engineering and deep learning. The process of feature engineering is to use the data to produce patterns that helps separate whether an event has occurred or has not occurred. These features are used to help localize an event using a *Z-score* function and are then be used as inputs to the ANN, which may provide improvements to event classification accuracy against providing the raw data into the ANN itself.

3.1.1 Critical Slowing Down

Research has observed that many statistically significant patterns that are indicative of power system related events. As mentioned above, CSD has been shown to predict some types of critical transitions on a power system [15, 22]. It has also been applied with machine learning to predict voltage collapse and instability in power grids [9, 33, 41]. Power system events typically occur during instability or cause critical transitions that destabilizes the grid, making CSD an ideal candidate for feature engineering. CSD is comprised of two features that together have been

shown to predict critical transitions:

1. Rolling autocorrelation of a detrended signal
2. Rolling variance of a detrended signal

3.1.1.1 Rolling Window Size

The rolling window size T is an important variable in CSD. Using a large T allows one to see gradual changes in autocorrelation and variance of a signal over a long time period, while a small T is more suitable for more rapid changes on a signal. Power grid events of interest here typically consist of relatively rapid changes, especially for equipment outages, so using a small T would be good for a PMU dataset.

3.1.1.2 Detrending the Signal

Detrending the signal removes the low frequency components of the signal, leaving behind only its high frequency components, which indicate times where fast changes occur in the signal. To detrend the signal, a Gaussian Kernel is applied to the data to capture the low pass frequency components of the data:

$$G(x, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \tag{3.1}$$

where x is the input data and σ is the standard deviation of the gaussian filter $G(\bullet, \bullet)$ [15].

The detrended data $y[k]$ can be found by subtracting the gaussian filtered signal from the original dataset $x[k]$.

$$y[k] = x[k] - G(x[k], \sigma) \tag{3.2}$$

3.1.1.3 Variance

The variance over a rolling window of size T given the detrended signal $y[k]$ can be found using the following:

$$Var[y[k]] = \frac{1}{T} \sum_{n=k-T}^{k} (y[n] - \mu_y)^2 \tag{3.3}$$

Where μ_y is the mean value of the detrended signal over the interval.

3.1.1.4 Autocorrelation

The autocorrelation coefficient a_c can be found by applying an autoregression function to $y[k]$ and then obtaining a_c by minimizing the following:

$$a_c = \arg\min_a \sum_{k=1}^{T} (y[k] - ay[k-1])^2 \tag{3.4}$$

3.1.2 Artificial Neural Network

Artificial Neural Networks have many functions, one of which is classifying data. However, there are many different architectures of ANNs and a general Convolutional Neural Network (CNN) may not be the best architecture to classify time-series data, as many better performing architectures of neural networks have been introduced. Our dataset has multiple PMUs, each with multiple features, so it is advised to apply 1-D convolutions to the dataset rather than 2 dimentional (2-D) convolutions. This is because features such as Voltage and Rate of Change of Frequency may be convoluted together. Additionally, 1 dimensional (1-D) max pooling should be used rather than 2-D max pooling as using 2-D max pooling may cause different features of PMUs to be pooled together. In this case the signal with a larger value, generally voltage, is always the selected pooling candidate.

In addition to needing 2-D operations to be changed to 1-D operations for the CNN, different architectures of CNNs may perform differently. In a 2018 research study, it was shown that both the ResNet architecture, and FCNs performed well with time series data, in contrast to Multilayer Perceptron models, Time-CNN models, and Multi-Scale Convolutional Neural Network (MCNN) models [27]. Due to the efficacy of both the ResNet and FCN models, both models are used here to classify events and are compared for accuracy.

3.1.2.1 Fully Convolutional Network

FCNs are CNNs with only convolutional layers, rather than having additional dense layers at the end of the network, excluding the final dense layer that classifies the data [32]. This can vastly improve model complexity due to the size of dense layers. This dense layer can be sacrificed in exchange for many more convolution layers, which in turn can provide more strictly convolutional feature maps. For long time-series data sets with multiple features, this can be useful as the time series data may either have to be heavily max-pooled along the time axis, which may reduce the resolution around the time of the event.

3.1.2.2 Residual Neural Networks

ResNets differ from CNNs by providing residual connections between convolution blocks in the network [25]. These residual connections help the optimization solver with identity mapping of features in deeper layers as it can draw on previous identity maps and solve for the perturbations of the residual identity maps rather than learning a new map. This has been proven to improve accuracy, especially in deeper layers.

3.1.3 Contributions

This chapter contributes to neural network based classification of event data by using critical slowing down to engineer new features out of the PMU data. These

features for both non-event PMU data and PMU event data is sent into a *Z-score* function, which helps localize and extract a predicted 6-second interval of data for the event or non-event. The extracted data is then input to a ResNet and a FCN and evaluated on classification accuracy. This classification accuracy is compared with using raw PMU data to localize and extract the predicted 6-second interval and are then be input into each network, where the raw PMU data is evaluated on classification accuracy. The contribution after analyzing the results from these experiments are determining whether feature engineering can localize events reliably, and if network architectures drastically change performance results for power-system related experiments.

3.2 Methodology

To test how well the Critical Slowing Down features perform, a set of four experiments are conducted, with network type and determining whether or not to use CSD being the variables between the experiment runs. The experiment descriptions are given below:

1. Critical Slowing Down applied to data. ResNet network used.
2. Critical Slowing Down applied to data. FCN network used.
3. Critical Slowing Down not applied to data. ResNet network used.
4. Critical Slowing Down not applied to data. FCN network used.

Using both the ResNet and the FCN provides robustness in determining how well the CSD features perform as it provides more insight as to whether the classification accuracy differences between CSD extracted features and the raw data are dependent on network architectures. Additionally, controlling for ResNet and FCN provides valuable comparisons in determining whether one architecture has a significant advantage over the other or not.

3.2.1 Network Architectures

In order to create a fair comparison between the networks, the FCN and ResNet networks are identical, other than the residual connections. They have the same depth, size, and dropout. The network architectures are depicted in Fig. 3.1.

3.2.2 Data and Data Localization

The data for this experiment is a set of real PMU data over a 2 year time period, with 30 Hz and 60 Hz sample rate PMUs. The data was be unstacked, providing a feature to the dataset for every PMU and PMU signal combination. The PMU signals include frequency, rate of change of frequency, voltage magnitude, and current magnitude in this experiment.

To account for the mismatch in PMU data rates, along with missing data due to PMU outages, linear interpolation was performed on the data to ensure all points have real values.

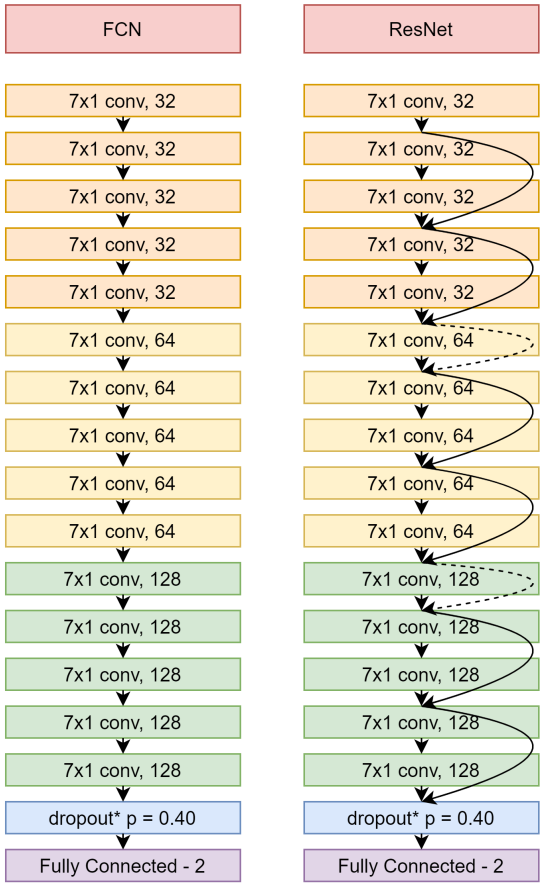

Figure 3.1: Network architectures of the FCN (left) and ResNet (right). Each network uses 3 blocks of 5 7x1 convolution layers. Each block has double the number of output filters with respect to the previous block.

Making a fair comparison to the input data is vital to the efficacy of the results of the experiment as well. Applying preprocessing methods to either the dataset with CSD applied to the dataset or the dataset without CSD applied and not the other will produce obscure results. Additionally, the events are given by the resolutions of minute intervals and it is observed that the events can happen between two minutes before the labeled event time and two minutes after the labeled event

Algorithm 1 CSD Preprocessing

Input: input dataset X
Output: output dataset Y
X = CSD(X)
X = X - X.mean("by_col") / X.std_dev("by_col")
ts = max(X.mean("by_row")).timestamp
Y = getSubset(X, ts-3seconds, ts+3seconds)
return Y

Algorithm 2 Raw Data Preprocessing

Input: input dataset X
Output: output dataset Y
X = X - X.mean("by_col") / X.std_dev("by_col")
ts = max(X.mean("by_row")).timestamp
Y = getSubset(X, ts-3seconds, ts+3seconds)
return Y

time. The use of 30 and 60 Hz PMU units results in approximately 14,400 samples of data over the four minute window where the event can occur. This will force the model to either be too shallow, too large computationally and take up too much memory, or require excessive max-pooling.

To accommodate these issues, a *Z-score* function was applied to the datasets to extract the timestamp with the highest average *Z-score* across all features, which is the predicted location of the event. This predicted location is used to extract the time range between 3 seconds before the timestamp with the highest average *Z-score* and the 3 seconds after the timestamp with highest average *Z-score*. The algorithms 1 and 2 show the preprocessing applied to both the CSD case and the non-CSD case. In each equation, the output dataset Y is the data that is input into the FCN and ResNet.

Because the localization may be prone to prediction errors, three data window sizes before extraction are investigated as separate experiments:

1. A 10 minute window, +/- 5 minutes from an event's labeled time.

2. A 6 minute window, +/- 3 minutes from an event's labeled time.

3. A 4 minute window, +/- 2 minutes from an event's labeled time.

These window sizes were chosen to provide insight as to whether CSD provides an improvement to the classification accuracy of the ANNs and/or if CSD provides improvements to the localization of events. As previously stated, there is an assumption in this particular dataset that events can happen up to two minutes before or after the labeled event time. These ranges were chosen such that events would not overlap with one another, and that the event actually occurred on the given interval.

3.2.3 Training the Network

As referenced in figure 3.1, a dropout of 0.4 is applied after the final convolutional layer to reduce overfitting on the dataset. The optimizer of the each network is the *Adagrad* optimizer, which has been found to perform well for sparse data. Considering each window is 6 seconds, and the processed data represents the CSD *Z-score* or the raw data *Z-score*, most of the data in the window is near zero and irrelevant to classification, while a small minority of the window contain the event, which likely have a much higher *Z-score* [43].

3.2.4 Hyperparameter tuning

To ensure that both the CSD case and non-CSD case are performing to their optimal condition, alongside the ResNet and FCN, hyperparameter tuning is applied using grid search to ensure the best learning rate and epoch is achieved. The learning rates applied to the hyperparameter tuning grid are the following: [1e-2, 5e-3, 1e-3, 5e-3, 1e-4, 5e-5]. The first 75 epochs were tested. All these tests were averaged using 4 sets distinct training and testing sets with a 25% and 75% respective split.

3.2.5 Evaluation of the Networks

The evaluation of the network consists of two metrics: the mean verification accuracy of 8 uniquely seeded sets of the data and the mean area under curve region of convergence (AUC-ROC) of those 8 uniquely seeded sets of data. The 8 sets were composed of 75% training data and 25% verification data sets for a more exhaustive overview of the data. The mean verification accuracy simply consists of the average verification accuracy of the 8 uniquely seeded sets. The mean area under curve region of convergence is a metric for confidence of the predictions. A higher AUC-ROC provides more confidence at distinguishing between events and non-events. Let the true positive rate and false positive rate be defined as follows:

$$TPR(k) = \int_{k}^{\infty} f_1(x)\,dx \tag{3.5}$$

$$FPR(k) = \int_{k}^{\infty} f_0(x)\,dx \tag{3.6}$$

where $f_1(x)$ is the conditional probability density function (PDF) of the event class score value given that the event class is the true class, $f_0(x)$ is the conditional PDF of the event class score value given that the non-event class is the true class, and k is a threshold parameter, to which the event class score is compared to make the classification decision. The AUC can then be calculated as follows:

$$AUC\_ROC = \int_{0}^{1} TPR(FPR^{-1}(x))\,dx = P(X_1 > X_0) \tag{3.7}$$

where X_1 is the score of an event, and X_0 is the score of a nonevent. Overall, this tells us the probability that an event score will be higher than a non-event score, which provides the probability that the event is classified correctly.

3.3 Results

The results of the FCN and ResNet comparing critical slowing down against non-critical slowing down are shown in Figures 2-7. Figures 2, 4, and 6 show the FCN cases with a +/- 2 minute interval, +/- 3 minute interval, and +/- 5 minute interval, respectively. Similarly, figures 3, 5, and 7 show the ResNet cases with a +/- 2 minute interval, +/- 3 minute interval, and +/- 5 minute interval, respectively.

The first important result from this experiment is the differences in network architecture. The median AUC-ROC's generally favor the ResNet over the FCN over the six experiments. However, the biggest difference comes into play when

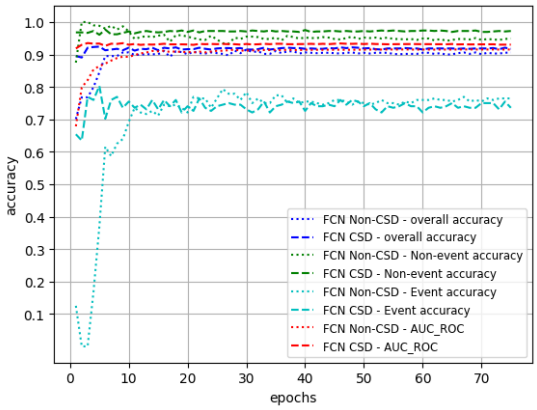

Figure 3.2: FCN Results, -2/+2 minutes from labeled event time - The results show that CSD improves the AUC-ROC over the raw data, and an improved verification accuracy, signifying better event localization.

applying CSD to the data. With CSD, the FCN provides slight improvements in the AUC-ROC, with verification accuracies similar to those of the raw data. However, the ResNet observes a large shift in performance as the event accuracy is much better with CSD than without CSD, while the non-event accuracy is worse with CSD than without CSD.

We observed that the dataset has a large number of potential critical transitions that were not labeled, which may indicate that there are many events that went unlabeled. For this reason, using the ResNet with CSD may be the cause of misidentifying many of the non-events as events, due to having properties of critical transitions (which may or may not be events). The results of all of these experiments can be summarized as Table 3.1, which includes the median AUC-ROC of an experiment x, denoted $\tilde{x}$, and the maximum AUC-ROC of an experiment x, denoted Max(x). The median was taken to represent the average over the

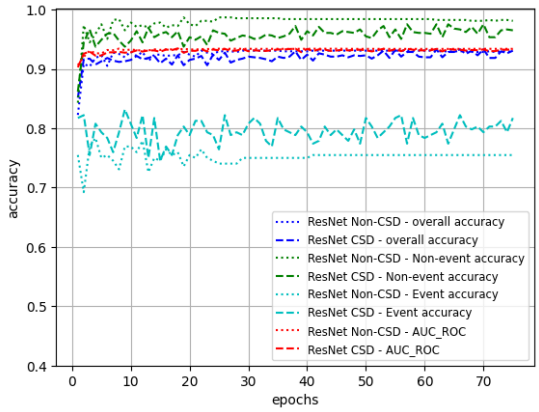

Figure 3.3: ResNet Results, -2/+2 minutes from labeled event time - The results show no significant improvement using CSD over the raw data

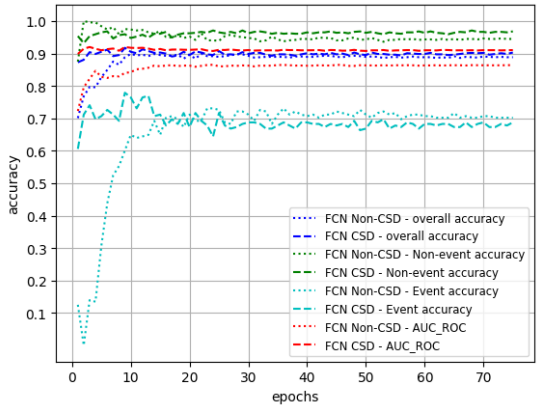

Figure 3.4: FCN Results, -3/+3 minutes from labeled event time - The results show that CSD improves the AUC-ROC over the raw data, signifying better event localization.

mean in consideration of the poor AUC-ROC during the first few epochs of the training. As one can see in Table 3.1, the maximum and median AUC-ROC using CSD outperformed the raw data for most test cases and at worst performed nearly identically to the raw data.

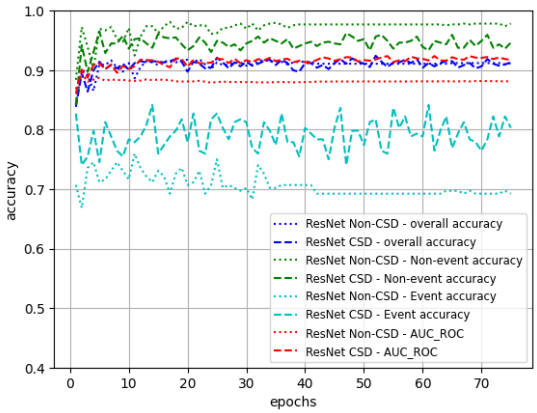

Figure 3.5: ResNet Results, -3/+3 minutes from labeled event time - The results show that CSD improves the AUC-ROC over the raw data, signifying better event localization.

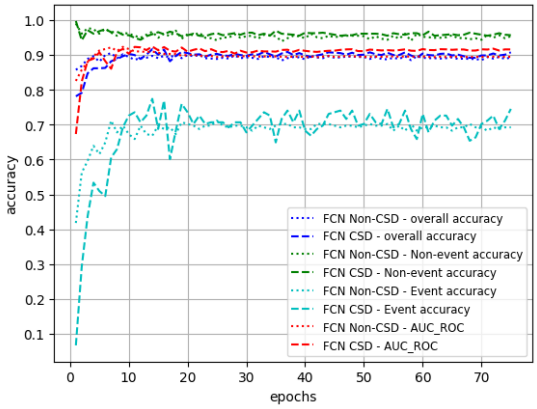

Figure 3.6: FCN Results, -5/+5 minutes from labeled event time - The results show that CSD improves the AUC-ROC over the raw data, signifying better event localization and classification.

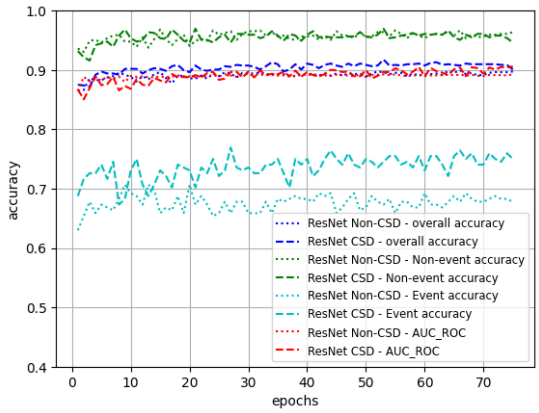

Figure 3.7: ResNet Results, -5/+5 minutes from labeled event time - The results show that CSD no significant improvement for the AUC-ROC over the raw data. However, it does show better verification accuracy.

Table 3.1: Performance comparison of the proposed CSD-enhanced ANNs by measuring AUC ROC

	$\tilde{x}$	Max(x)
FCN Raw Data (+/- 2 min)	0.9141	0.9141
FCN CSD (+/- 2 min)	0.9317	0.9351
ResNet Raw Data (+/- 2 min)	0.9338	0.9342
ResNet CSD (+/- 2 min)	0.9313	0.9346
FCN Raw Data (+/- 3 min)	0.8626	0.8651
FCN CSD (+/- 3 min)	0.9101	0.9199
ResNet Raw Data (+/- 3 min)	0.8809	0.8943
ResNet CSD (+/- 3 min)	0.9165	0.9236
FCN Raw Data (+/- 5 min)	0.8951	0.9229
FCN CSD (+/- 5 min)	0.9125	0.9240
ResNet Raw Data (+/- 5 min)	0.8906	0.8928
ResNet CSD (+/- 5 min)	0.8926	0.9057

3.4 Conclusion and Future Work

Protection of the power grid is an important function in today's society, which depends on electrical power more than it ever has before. Being able to acknowledge when critical events occur on the grid allows us to more accurately perform risk analysis, closely monitoring the power flow and wide-area stability after events and during system recovery. We took real PMU data and engineered new features using the CSD technique. These features were fed into residual and FCN networks and the classification accuracies of events against non-events were calculated. The raw data was also fed into these neural networks as a baseline comparison.

The results consistently showed that the CSD engineered features had higher AUC-ROC overall than the raw data. Although the verification accuracy was similar for both cases, the AUC-ROC suggests that CSD provides more confidence in the correct decisions, or less confidence in the incorrect decisions, both of which make expected accuracy to improve with more data. Additionally, the ResNet improved the AUC-ROC further. Finally, when CSD was applied with the ResNet, the event accuracy improved and the non-event accuracy dropped. Due to the presence of unlabeled critical transitions in the data, there may be events in the non-event data, which could only be captured using the ResNet and CSD.

Future work can improve upon this research by analyzing more feature engineering patterns that may show significant results on all types of events and combining this with the Critical Slowing Down method. Additionally, formulating a data augmentation onto the events or synthesizing more event data could provide

tremendous value, so the network can more easily learn the significant features of the events, while also being less prone to overfitting. Synthesizing PMU data for events using power system models could be critical in ensuring that the network comprehensively learns an event's features. Additionally, using some combination of synthesized events and real events, new events can be synthesized using state of the art Generative Adversarial Networks [55, 60]. Finally, we also plan to explore other network architectures to possibly improve performance and training time for the data, such as the Transformer network architecture [51].

Chapter 4: Performance Scaling of Algorithms: Evaluation of GPU vs. CPU Architectures for Time-domain Power System Data

PMUs continue to be instrumental in determining whether or not events have occurred on the grid, and erroneous interpretation by machine learning algorithms could possibly causing misleading grid stability verification results. These inaccuracies could increase the probability that another high magnitude blackout could occur, drastically impacting the economy and people's lives. Critical Slowing Down, alongside many of the other aforementioned machine learning algorithms, are used on the grid to isolate when the grid is unstable or whether there has been an equipment outage on the grid.

One of the challenges with PMUs is the amount of data that is being accumulated. Most PMUs operate at 30 to 60 Hz, which is dramatically higher than previously used supervisory control and data acquisition systems (SCADA) [1]. SCADA operates at a slower sample rates of 0.1 to 0.2 Hz, such that its algorithms do not need to be designed for enormous datasets and high sample frequencies [36]. PMU-based algorithms acquire huge datasets and must operate at high frequencies. To achieve this, the algorithms must be computationally efficient and scale well to large datasets.

One important factor in how fast the algorithm can run is the type of computer architecture the algorithm is running on. It has been noted that GPU computing

works well on computationally complex problems and can dramatically decrease computation time for algorithms where parallelism is applicable [23]. However, it has also been noted that for many iterative algorithms, sequential implementations of algorithms outperform their GPU enabled counterparts [14]. However, many machine learning algorithms can benefit the GPU due to benefiting from parallelism and being computationally complex, while also being iterative. This complicates the decision of whether to use a GPU implementation of an algorithm or a sequential CPU implementation of an algorithm.

This chapter will focus on benchmarking GPU implementations and CPU implementations of common machine learning algorithms used in PMU data analytics to provide a reference of what architecture scales best for for PMU data analytic algorithms.

4.1 Clustering Algorithms for PMU data Analytics

Clustering algorithms are used in PMU data Analytics for many reasons, such as anomaly detection, partitioning oscillatory modes, power system network partitioning, and data visualization [20, 30, 37, 53, 62]. In specific, K-means clustering and Density-based spatial clustering of applications with noise (DBSCAN) are two widely used clustering algorithms for PMU data.

K-means works by specifying K clusters that the data will be fit to. The

objective function is to minimize the following function:

$$f(S, x) = \min_{S} \sum_{i=1}^{k} \sum_{x \in S_i} \ell_2(x, \mu_i) \tag{4.1}$$

Where S_i is the set of elements belonging to cluster i, μ_i is the mean of cluster set S_i, S is the superset of all sets, and ℓ_2 is the L2-norm function. An abstract algorithm for this can be given as follows:

Algorithm 3 K-means algorithm

Randomly select k datapoints from X to initialize centroids of the k clusters
while Improvements are made **do**
  Assign each datapoint in X to S_i with the nearest centroid μ_i
  Recompute the μ_i for all S_i.
end while

DBSCAN on the other hand is a more complex algorithm that has a dynamic number of clusters. DBSCAN requires a distance threshold ϵ and a minimum number of points for cluster k. As the name suggests, DBSCAN creates clusters based on density. There are two different types of points in a cluster: core points, and border points. Core points are points that have at least k neighboring points within the distance radius ϵ, while border points have less than k neighboring points within that radius [18]. A point p is directly density-reachable of a point q if p is in the ϵ radius of q and q is a core point. Finally, a point q is density-reachable to p if there exists a chain of points $q, x_1, x_2, \ldots, x_n, p$, such that each pair x_{i-1}, x_i are directly density reachable [18].

Using core points, direct density-reachability, and density-reachability, the al-

gorithm for DBSCAN can be given as follows:

Algorithm 4 DBSCAN abstract algorithm

Compute the number of neighbors for each point p in X
Classify each point p with at least k neighbors within ϵ as core points.
Merge all density-reachable core points together in their own clusters.
Assign each non-core point to either a cluster if it is within ϵ of a core point.
Otherwise, label it as noise.

To compare the runtime and scaling of the GPU algorithm and CPU algorithms for K-means and DBSCAN, two experiments will be conducted for each algorithm. The first experiment will hold the number of features in the dataset constant and change the number of datapoints. In this experiment the number of datapoints will be held constant at 25000. The second experiment will hold the number of datapoints constant and alter the number of features. The number of features will be held to 64. The algorithms that will be looked at use the RAPIDS AI CUML library for the Cuda-GPU implementation of the algorithms and Scikit-Learn for the CPU implementations of the algorithms [39, 49].

4.1.1 Results

Figure 4.1 shows that the GPU implementation scales better with an increased number of features as opposed to the CPU implementation. The CPU implementation has a much more significant positive trend for computation time vs. number of features than the GPU implementation, likely due to the GPU being able to parallelize the L2-norm function across features. Figure 4.2 also shows that the

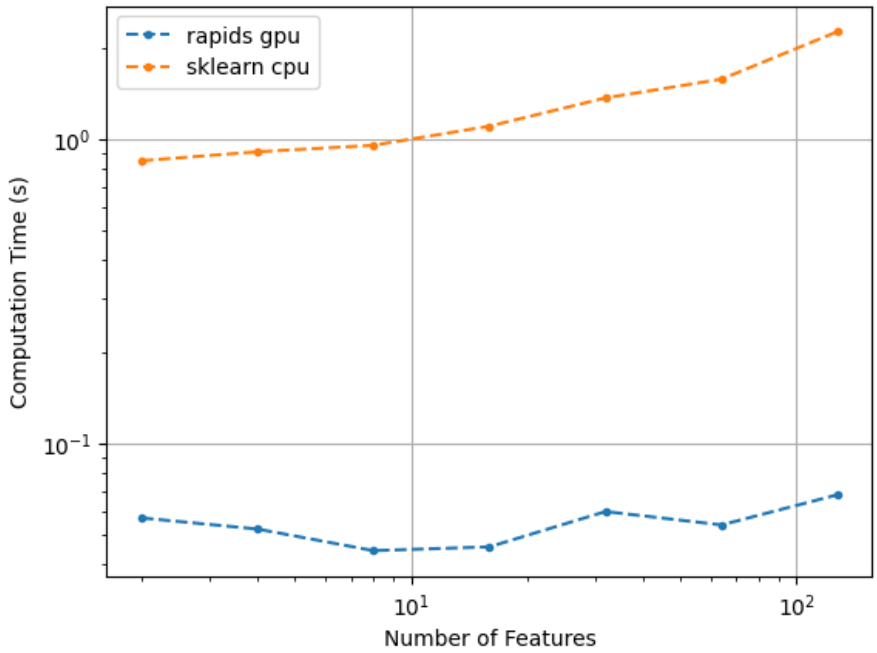

Figure 4.1: Kmeans: Number of Features vs. Computation Time. Number of datapoints constant at 25000

GPU implementation outperforms the CPU implementation of Kmeans. As the number of datapoints increase, the GPU implementation appears to outperform the CPU slightly for a small number of datapoints, and then significantly for a larger number of datapoints.

Similar to Kmeans, Figure 4.3 shows that the GPU DBSCAN implementation scales better with an increased number of features as opposed to the CPU implementation. The difference between the CPU implementation and the GPU implementation appears more significant for the DBSCAN implementation, likely due to the complexity of the algorithm. Figure 4.4 shows that the GPU implementation takes longer for a small number of datapoints, and then begins to outperform

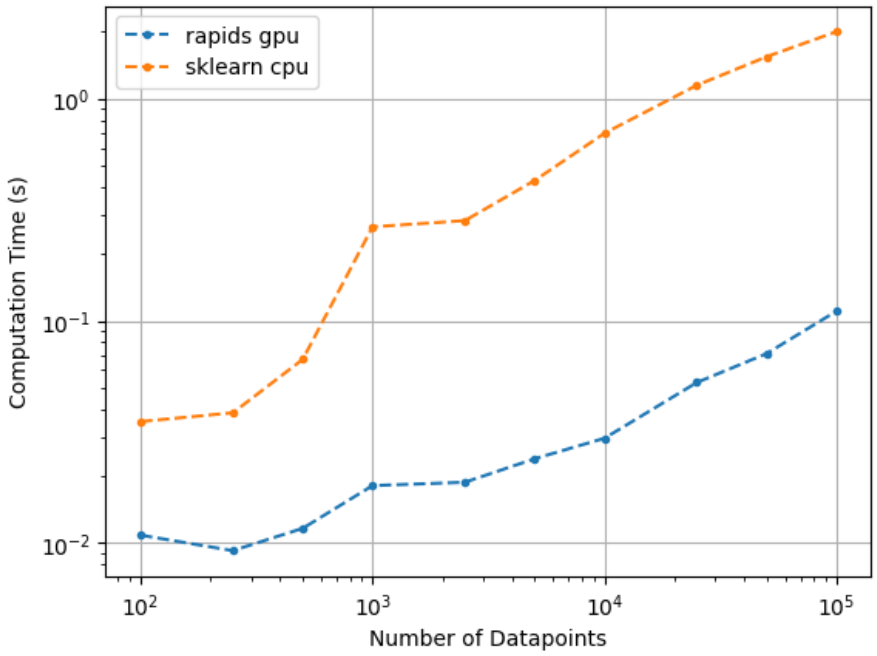

Figure 4.2: Kmeans: Number of Datapoints vs. Computation Time. Number of features constant at 64

the CPU implementation.

Based on this experiment, it becomes clear that if there is a large number of features for the PMU algorithm (which can consist of all reported data for multiple PMUs), the GPU implementations of DBSCAN and K-means will outperform the CPU implementation counterparts. For a high number of datapoints, the GPU implementation may also be useful to a lesser extent than a high number of features.

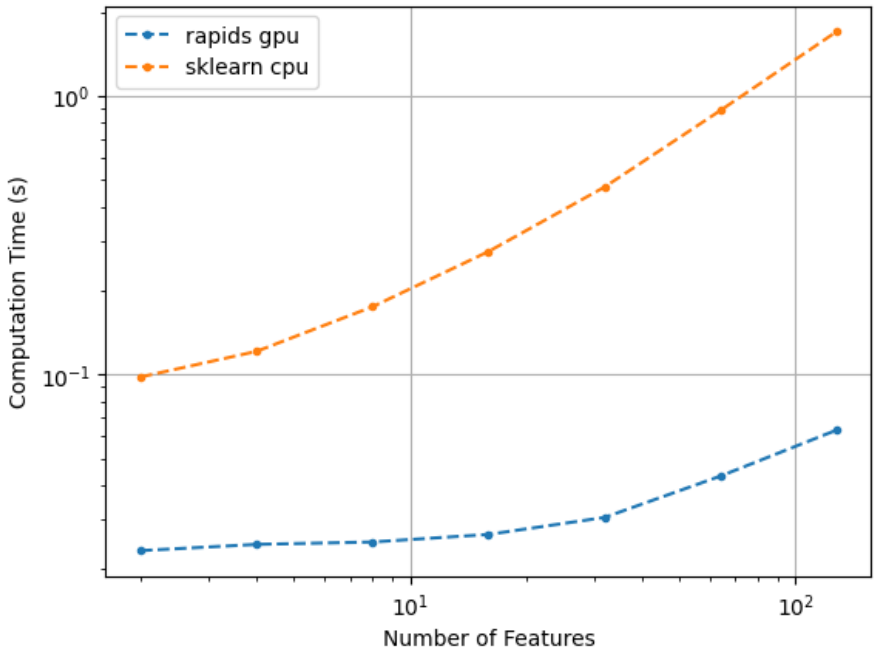

Figure 4.3: DBSCAN: Number of Features vs. Computation Time. Number of datapoints constant at 25000

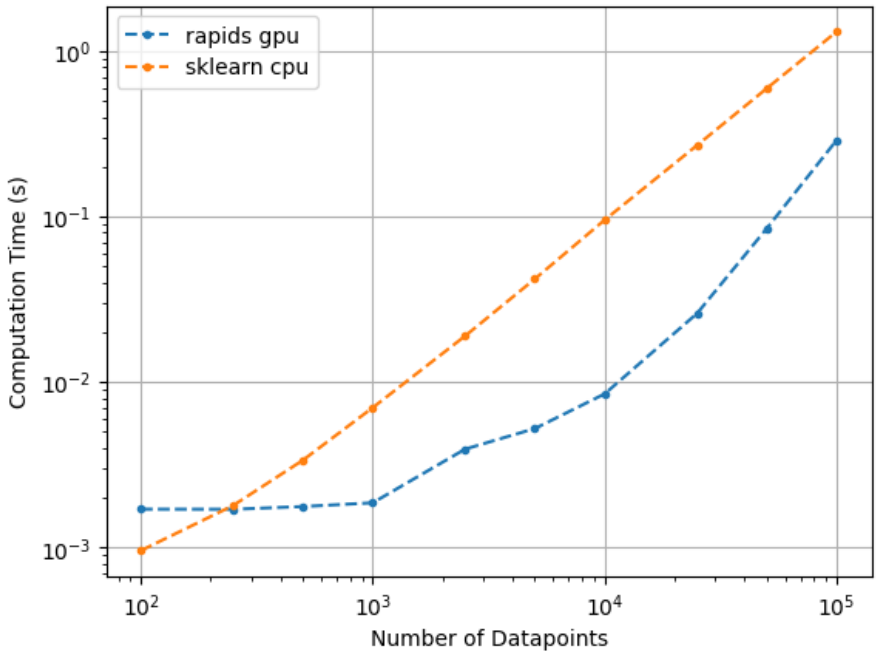

Figure 4.4: Kmeans: Number of Datapoints vs. Computation Time. Number of features constant at 64

4.2 Rolling Window Applications and Critical Slowing Down

Rolling window applications, such as the rolling autocorrelation and rolling variance used in CSD has been used in PMU data analytics for anomaly detection and as precursory indicators for instability and events in the grid [33, 41]. One of the important notes about rolling window calculations is that the solutions can often be iteratively updated, such that a the rolling window of elements $X_{n-W+1}, \ldots, X_n$ can be computed using X_n, X_{n-W}, and the results of the rolling window calculations of elements $X_{n-W}, \ldots, X_{n-1}$, where W is the window size. This is much more efficient than computing variance, autocorrelation, or some other statistic on each window of size W.

The rolling variance can be updated iteratively using the mean of the previous W elements and the element X_n that is replacing X_{n-W}. The equation for the rolling update is given as follows:

$$\begin{aligned} \mu_{update} &= \mu_{prev} + \frac{X_n - X_{n-W}}{W} \\ \sigma^2_{update} &= (X_n - X_{n-W})(X_n - \mu_{update} + X_{n-W} - \mu_{prev}) \end{aligned} \tag{4.2}$$

The autocorrelation function is simply calculating the pearson coefficient between a dataset X and a lagged copy of dataset X, X_{lag}. The pearson coefficient can easily be broken up into a sum of $X \cdot X_{lag}$ and the square root of the variance of X and X_{lag}. Thus, we can easily break rolling autocorrelation into simpler problems of finding the rolling sum of $X \cdot X_{lag}$ and the variances of X and X_{lag} and then combining them.

As can be seen by the rolling variance formula (and consequently the rolling autocorrelation formula that follows from it), rolling window statistical calculations can often be simplified into non-complex iterative update algorithms, which should benefit the sequential CPU implementation.

For the rolling window calculations for both autocorrelation and variance, the only experiment will look at the number of features vs computation time. The window size does not matter as the update formula only depends on values of the previous calculation and the new and replaced datapoints. Regardless of the window size, the iterative update can be done in constant time.

4.3 Results

Figure 4.5 shows the results for the rolling variance calculation. As the number of features increases, the GPU implementation of rolling variance begins to approach the CPU implementation. However, the CPU implementation outperforms the GPU implementation for at least up to 512 features. The rolling variance implementation is better suited for CPU unless mass parallelization can be utilized.

Figure 4.6 shows similar results for the rolling autocorrelation calculation. Once again, the rolling autocorrelation performs much better on the CPU at a low number of features. However, at 128 features, the GPU implementation begins to outperform the CPU implementation. The rolling autocorrelation update is a more complex update than the rolling variance update, which explains why the GPU gains more significant advantage on the rolling autocorrelation over rolling

variance. However, the update for rolling autocorrelation is still not complex, which suggests that rolling autocorrelation should be implemented on the CPU unless mass parallelization can be utilized.

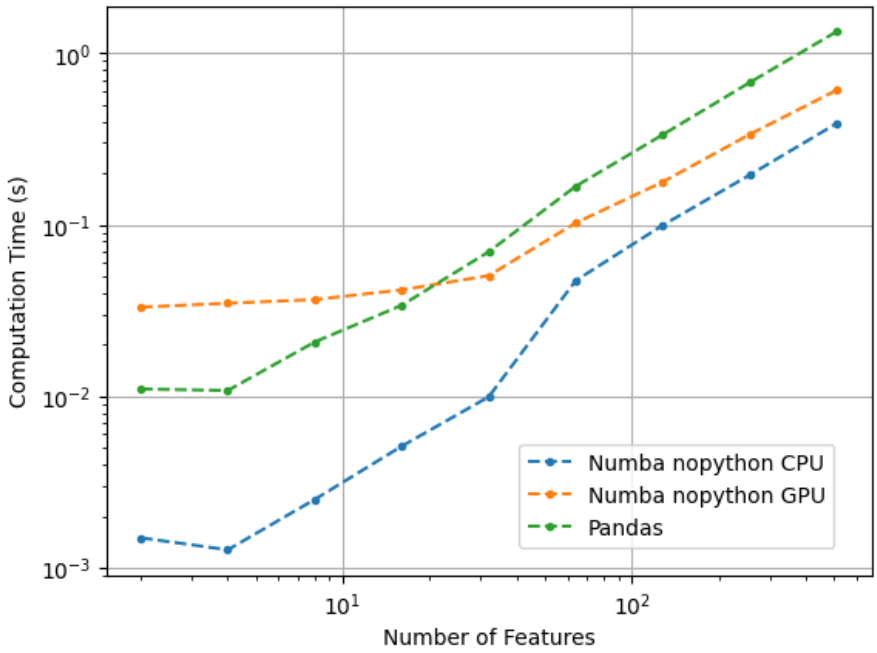

Figure 4.5: Rolling Variance: Number of Features vs. Computation Time

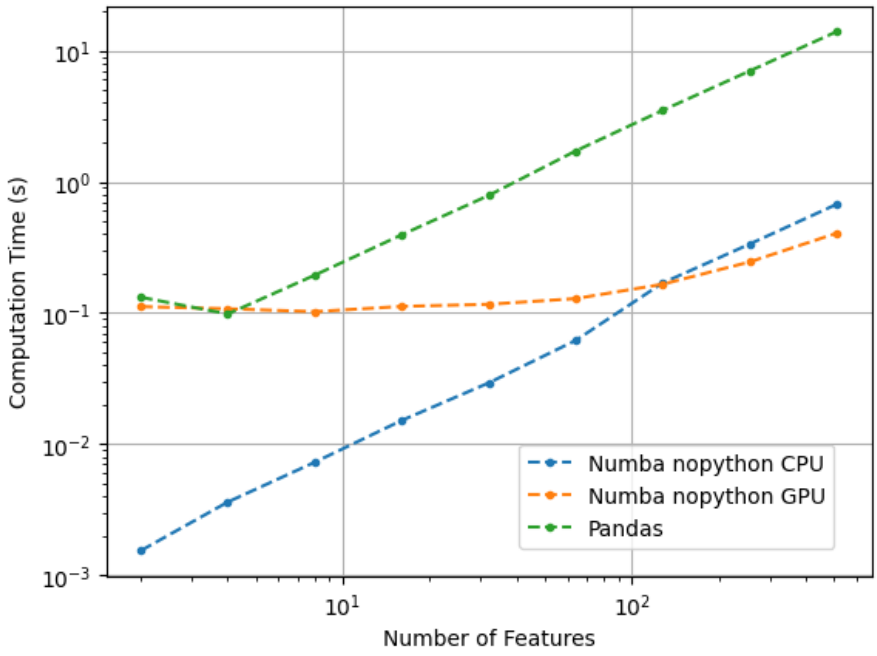

Figure 4.6: Rolling Autocorrelation: Number of Features vs. Computation Time

4.4 Principal Component Analysis

Principal Component Analysis (PCA) is a widely used technique that leverages PMU data. It has been used in applications such as PMU spoofing error detection, fault location, outlier detection, and many others [17,34,58]. PCA can be computed using two techniques: Singular Value Decomposition (SVD) of the data X or the Eigendecomposition of the $X's$ covariance matrix.

SVD decomposes the data matrix X as follows:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{W}^T \tag{4.3}$$

Where $\boldsymbol{U} \in \boldsymbol{R}^{n \times n}$ contains column vectors that are orthogonal unit vectors, $\boldsymbol{W} \in \boldsymbol{R}^{p \times p}$ contains column vectors that are orthogonal unit vectors, and $\boldsymbol{\Sigma} \in \boldsymbol{R}^{n \times p}$ is a diagonal matrix containing the singular values σ_k across the diagonal. SVD can then retrieve a truncated version of the $\boldsymbol{X}$ by considering the first k columns of vector $\boldsymbol{U}$, denoted as $\boldsymbol{U}_k$, the first k columns and rows of $\boldsymbol{\Sigma}$, denoted by $\boldsymbol{\Sigma}_k$, and the first k rows of $\boldsymbol{W}^T$, denoted as $\boldsymbol{W}_k^T$:

$$\tilde{\boldsymbol{X}} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{W}_k^T \tag{4.4}$$

Method 2 for computing PCA includes computing the covariance matrix of X as follows:

$$\boldsymbol{K_{XX}} = \boldsymbol{E}[\boldsymbol{X}\boldsymbol{X}^T - \mu_{\boldsymbol{X}} \mu_{\boldsymbol{X}^T}] \tag{4.5}$$

$\boldsymbol{K_{XX}}$ can then be decomposed as follows:

$$\boldsymbol{K_{XX}} = \boldsymbol{Q} \boldsymbol{\Lambda} \boldsymbol{Q}^T \tag{4.6}$$

Where $\Lambda \in \boldsymbol{R}^{n \times n}$ is a diagonal matrix containing the eigenvalues of $\boldsymbol{K_{XX}}$ and $\boldsymbol{Q}$ is an orthogonal matrix containing the eigenvectors of $\boldsymbol{K_{XX}}$. Like the SVD method, $\boldsymbol{X}$ can be truncated by the following:

$$\tilde{\boldsymbol{X}} = \boldsymbol{Q}_k \boldsymbol{\Lambda}_k \boldsymbol{Q}_k^T \tag{4.7}$$

The experiments for PCA will compare both of these methods. Two experi-

ments will be conducted. The first experiment will vary the number of features (columns) and hold the number of datapoints (rows) constant at 25000. The second experiment will vary the number of datapoitns (rows) and hold the number of features constant at 128. Each reduction will reduce the number of features from p to $\lfloor \frac{p}{2} \rfloor$.

4.4.1 Results

The results of both experiments can be shown in Figure 4.7 and Figure 4.8. Both experiments show similar trends. The covariance method outperforms the SVD method for this specific dataset. The SVD approach with the GPU scales poorly due to memory consumption and begins to fail at a large number of datapoints. On the other hand, for the covariance and eigendecomposition approach, the CPU implementation is favored for small datasets. As the number of features or the number of datapoints rises, the GPU implementation begins to outperform the CPU implementation due to the complexity of the calculation. PMU data will generally only have less than 10000 datapoints if calculating on less than 3 minutes of data, so the GPU implementation would likely be preferred as memory allows and depending on CPU and GPU specifications.

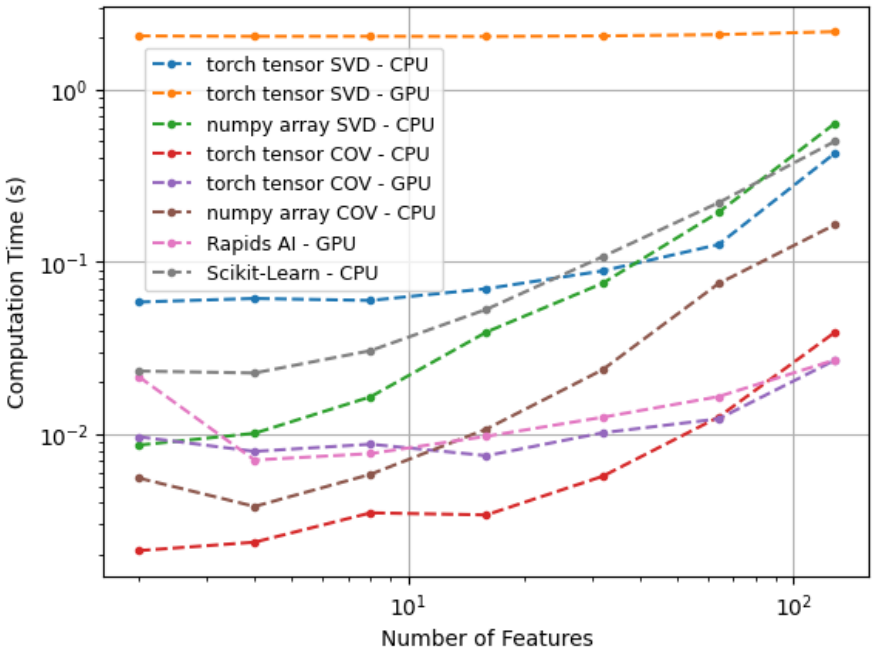

Figure 4.7: PCA: Number of Features vs. Computation Time

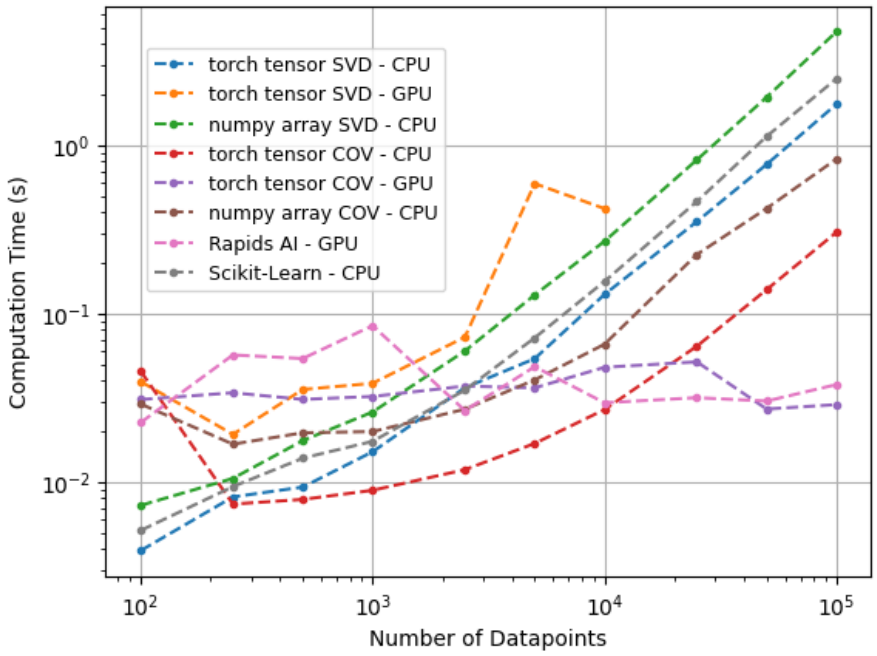

Figure 4.8: PCA: Number of Datapoints vs. Computation Time

Chapter 5: Conclusions and Future Work

In this Chapter we summarize our findings and discuss future steps to build upon the research in this Thesis toward facilitating the use of time-domain power system data in artificial intelligence approaches for the grid.

5.1 Conclusions

As global trends continue to consume more electricity for various reasons, our lives and the economy become more reliant on electricity, thus causing need for a more reliable electric grid. Instabilities in the electric grid can cause unstable grid condition, causing the grid to operate at far-from-optimal frequencies, currents, and voltages. In some situations, this can damage critical electronic equipment, placing the grid at a greater risk. These problems can often lead to load shedding, which can be a major inconvenience for the people effected. While the grid in an unstable condition, there is a higher risk that an event such as a line outage will overload other equipment, possibly leading to a cascading failure. One such of these cascading failures was the Northeast US blackout that cost over $5 billion to the US economy. With the rise of electric vehicles, ecommerce, and the economy becoming more reliant on electricity to operate, a blackout of a similar magnitude would be catastrophic. This chapter has looked at common problems of evaluating

threats to our grid and has introduced new ways into addressing these common problems. Doing so will provide a more accurate evaluation of the power grid's state, allowing utilities and operators to mitigate potential threats to our grid in a preventative manner.

In Chapter 2, we introduced CA and discussed that it did not account for aging within the grid. Excessive aging of electrical equipment can cause the equipment to fail below the rated power levels provided. For example, transmission lines lose tensile strength over time, increasing the risk of the tensile breaking under tension. Transformers lose their bushing insulation over time, which will increase the likelihood of a fault. Similarly, Generators can be placed under high mechanical and electrical stress, which can also cause their insulation to wear away over time. To account for these issues, chapter 2 introduced exponential aging models for transformers, transmission lines, and generators. These aging models were modeled onto the grid by decreasing the power rating of equipment as each piece of equipment were aged. Using AC-OPF, we verified that the number of contingencies increase as the average equipment age was increased on the grid, suggesting that current CA may underestimate the number of contingencies on the grid. More work can be done to refine the implementation of aging models onto the grid, along with improving the accuracy of the equipment models.

Chapter 3 introduced PMUs, which provide real-time data of the grid. One use of the PMU data is using machine learning and artificial intelligence algorithms to determine when an event takes place on the grid, such as a line outage or transformer outage. It is critical to determine if there is a outage on the grid, as the

results of CA assumes all components of the grid is operating, unless told otherwise. Chapter 3 introduced CSD, a statistical method applied to a rolling window, that is used as preprocessing to the data to input into ANNs. The experiment found that the maximum and median AUC-ROC using CSD outperformed the raw data for most cases and at worst performed nearly identically to the CSD applied data.

One other problem that causes issues in grid reliability and the efficacy of machine learning algorithms is the computational complexity of the algorithm and how fast it can run on the PMU data. An algorithm that is designed to detect anomalies is worthless if it is unable to classify incoming data in a timely manner, as the results of the algorithm will lag behind when the event took place. This will lead to the event not being resolved in a timely manner, leading to reactionary solutions rather than preventative solutions. Chapter 4 introduces GPU computing in which it looked at many commonly used algorithms that are used with PMU data and benchmarked performance of a GPU implemented algorithm against its CPU implemented counterpart. Chapter 4 found that computationally complex algorithms, such as clustering algorithms, scale well when using a GPU. More iterative and less complex algorithms, such as rolling window applications, generally benefit more from using a CPU, unless mass parallelization along features is applicable. Finally, PCA benefits heavily from using a GPU if the dataset is large in both number of features and number of datapoints. In general, GPUs scale better for these algorithms for larger datasets and more complex computational steps in the algorithm.

5.2 Future Work

Many advancements in the fields of artificial intelligence, algorithms, and data structures are important areas for exploration in power systems.

Further research toward detecting events on the grid is critical to ensure utilities and operators can work toward counteracting the events to restabilize the grid. Explore other network architectures to possibly improve performance and training time for the data, such as the Transformer network architecture and various graphical neural networks [51, 61].

Research in reinforcement learning is gaining traction for emergency control and power management [26, 57]. Further research can look at other aspects of the grid that can benefit from the introduction of reinforcement learning. Reinforcement learning can also optimistically look at whether PMU data and power system algorithm outputs can provide enough information to control an entire grid. Successfully doing so could show that reinforcement learning is capable of looking at the current state of the grid, and making experienced decisions, such as changing current generation or shedding specific loads.

The grid is also a sparse network, such that buses are only connected to a small subset of the other buses in the network. Many numerical algorithms and machine learning techniques use matrix computations on the adjacency matrices and other sparse matrices provided by the power system network. As a result, these matrix computations can be highly inefficient as only a small proportion of the computations are necessary. Techniques have utilized sparse matrices to

improve computations on power flow algorithms, such as the Newton-Raphson power flow method [48]. Many other algorithms can be computationally improved from sparse matrix evaluation. Power system algorithms can also benefit from improvements in sparse matrix data structures and algorithms.

Bibliography

[1] Ieee standard for synchrophasor measurements for power systems. *IEEE Std C37.118.1-2011 (Revision of IEEE Std C37.118-2005)*, pages 1–61, 2011.

[2] G. Andersson, P. Donalek, R. Farmer, N. Hatziargyriou, I. Kamwa, P. Kundur, N. Martins, J. Paserba, P. Pourbeik, J. Sanchez-Gasca, R. Schulz, A. Stankovic, C. Taylor, and V. Vittal. Causes of the 2003 major grid blackouts in north america and europe, and recommended means to improve system dynamic performance. *IEEE Transactions on Power Systems*, 20(4):1922–1928, 2005.

[3] Mohammad Arshad, S.M. Islam, and Abdul Khaliq. Power transformer aging and life extension. pages 498 – 501, 10 2004.

[4] D. Baimel, S. Tapuchi, and N. Baimel. Smart grid communication technologies- overview, research challenges and opportunities. In *2016 International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*, pages 116–120, 2016.

[5] Nouruddeen Bashir, Hussein Ahmad, and Mohd Shaffuan Suddin. Ageing studies on transmission line glass insulators using dielectric dissipation factor test. *2010 Conference Proceedings IPEC*, pages 1062–1066, 2010.

[6] N. Bhatnagar and B. Venkatesh. Energy storage and power systems. In *2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4, 2012.

[7] M. M. I. Bhuiyan, P. Musilek, J. Heckenbergerova, and D. Koval. Evaluating thermal aging characteristics of electric power transmission lines. In *CCECE 2010*, pages 1–4, 2010.

[8] M. H. J. Bollen. Effects of adverse weather and aging on power system reliability. *IEEE Transactions on Industry Applications*, 37(2):452–457, 2001.

[9] Samuel C. Chevalier and Paul D. H. Hines. Mitigating the risk of voltage collapse using statistical measures from pmu data. *IEEE Transactions on Power Systems*, 34(1):120–128, 2019.

[10] E. Ciapessoni, D. Cirio, A. Pitto, G. Kjølle, S. H. Jakobsen, and M. Sforna. Contingency screening starting from probabilistic models of hazards and component vulnerabilities. In *2016 Power Systems Computation Conference (PSCC)*, pages 1–8, 2016.

[11] L. A. Clarfeld, M. J. Eppstein, P. D. H. Hines, and E. M. Hernandez. Assessing risk from cascading blackouts given correlated component failures. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7, 2018.

[12] Federal Energy Regulatory Commission. Ferc order 888, 1996.

[13] Federal Energy Regulatory Commission. Ferc order 2000, 2000.

[14] Elise Cormie-Bowins. A comparison of sequential and gpu implementations of iterative methods to compute reachability probabilities. *Electronic Proceedings in Theoretical Computer Science*, 99:20–34, Oct 2012.

[15] E. Cotilla-Sanchez, P. D. H. Hines, and C. M. Danforth. Predicting critical transitions from time series synchrophasor data. *IEEE Transactions on Smart Grid*, 3(4):1832–1840, 2012.

[16] Mingjian Cui, Jianhui Wang, Jin Tan, Anthony R. Florita, and Yingchen Zhang. A novel event detection method using pmu data with high precision. *IEEE Transactions on Power Systems*, 34(1):454–466, 2019.

[17] Shashini De Silva, Jinsub Kim, and Eduardo Cotilla-Sanchez. Data driven sparse error correction for pmu measurements under gps spoofing attacks. In *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, 2021.

[18] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press, 1996.

[19] N. Fan, R. Chen, and J. Watson. N-1-1 contingency-constrained optimal power flow by interdiction methods. In *2012 IEEE Power and Energy Society General Meeting*, pages 1–6, 2012.

[20] Mohammadreza Maddipour Farrokhifard, Mohammadreza Hatami, Vaithianathan Mani Venkatasubramanian, Gilles Torresan, Patrick Panciatici, and

Florent Xavier. Clustering of power system oscillatory modes using dbscan technique. In *2019 North American Power Symposium (NAPS)*, pages 1–6, 2019.

[21] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel. Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions. *IEEE Transactions on Power Systems*, 28(4):4909–4917, 2013.

[22] G. Ghanavati, P. D. H. Hines, T. I. Lakoba, and E. Cotilla-Sanchez. Understanding early indicators of critical transitions in power systems from autocorrelation functions. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 61(9):2747–2760, 2014.

[23] Jayshree Ghorpade. Gpgpu processing in cuda architecture. *Advanced Computing: An International Journal*, 3(1):105–120, Jan 2012.

[24] J. F. Hall and A. K. Deb. Prediction of overhead transmission line ampacity by stochastic and deterministic models. *IEEE Transactions on Power Delivery*, 3(2):789–800, 1988.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[26] Qiuhua Huang, Renke Huang, Weituo Hao, Jie Tan, Rui Fan, and Zhenyu Huang. Adaptive power system emergency control using deep reinforcement learning. *IEEE Transactions on Smart Grid*, 11(2):1171–1182, 2020.

[27] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, Mar 2019.

[28] Cédric Josz, Stéphane Fliscounakis, Jean Maeght, and Patrick Panciatici. Ac power flow data in matpower and qcqp format: itesla, rte snapshots, and pegase, 2016.

[29] Sadegh Kamali and Turaj Amraee. Prediction of unplanned islanding in power systems using pmu data. In *2018 IEEE International Conference on Environment and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, pages 1–5, 2018.

[30] Ehdieh Khaledian, Shikhar Pandey, Pratim Kundu, and Anurag K. Srivastava. Real-time synchrophasor data anomaly detection and classification using isolation forest, kmeans, and loop. *IEEE Transactions on Smart Grid*, 12(3):2378–2388, 2021.

[31] D. Kourounis, A. Fuchs, and O. Schenk. Toward the next generation of multiperiod optimal power flow solvers. *IEEE Transactions on Power Systems*, 33(4):4005–4014, 2018.

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.

[33] J. Ma, J. Tang, Z. Yan, F. Jiang, H. Zeng, and C. Fang. Data-driven power system collapse predicting using critical slowing down indicators. In *2018 International Conference on Power System Technology (POWERCON)*, pages 1879–1884, 2018.

[34] Kaveri Mahapatra, Nilanjan Ray Chaudhuri, and Rajesh Kavasseri. Online bad data outlier detection in pmu measurements using pca feature-driven ann classifier. In *2017 IEEE Power Energy Society General Meeting*, pages 1–5, 2017.

[35] V. J. Mishra and M. D. Khardenvis. Contingency analysis of power system. In *2012 IEEE Students' Conference on Electrical, Electronics and Computer Science*, pages 1–4, 2012.

[36] Biswajit Mondal, Amit Kumar Choudhury, M Viswanadh, S P Barnwal, and D K Jain. Application of pmu and scada data for estimation of source of forced oscillation. In *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, pages 1–7, 2019.

[37] Anupam Mukherjee, Ranganath Vallakati, Valentin Lachenaud, and Prakash Ranganathan. Using phasor data for visualization and data mining in smartgrid applications. In *2015 IEEE First International Conference on DC Microgrids (ICDCM)*, pages 13–18, 2015.

[38] New York Independent System Operator. New york's clean energy grid of the future. *The New York ISO Annual Grid & Markets Report*, page 12, 2021.

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

[40] S. D. Picard, M. G. Adamiak, and V. Madani. Fault location using pmu measurements and wide-area infrastructure. In *2015 68th Annual Conference for Protective Relay Engineers*, pages 272–277, 2015.

[41] D. Podolsky and K. Turitsyn. Critical slowing-down as indicator of approach to the loss of stability. In *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 19–24, 2014.

[42] V. Roy, S. S. Noureen, S. B. Bayne, A. Bilbao, and M. Giesselmann. Event detection from pmu generated big data using r programming. In *2018 IEEE Conference on Technologies for Sustainability (SusTech)*, pages 1–6, 2018.

[43] Sebastian Ruder. An overview of gradient descent optimization algorithms, 2017.

[44] Bishal Silwal and Peter Sergeant. Thermally induced mechanical stress in the stator windings of electrical machines. *Energies*, 11(8), 2018.

[45] P. Smith and A. Schaeffer-Filho. Management patterns for smart grid resilience. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pages 415–416, 2014.

[46] M. B. Srinivas and T. S. Ramu. Multifactor aging of hv generator stator insulation including mechanical vibrations. *IEEE Transactions on Electrical Insulation*, 27(5):1009–1021, 1992.

[47] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp, 2019.

[48] Xueneng Su, Chuan He, Tianqi Liu, and Lei Wu. Full parallel power flow solution: A gpu-cpu-based vectorization parallelization and sparse techniques for newton–raphson implementation. *IEEE Transactions on Smart Grid*, 11(3):1833–1844, 2020.

[49] RAPIDS Development Team. *RAPIDS: Collection of Libraries for End to End GPU Data Science*, 2018.

[50] K. S. Turitsyn and P. A. Kaplunovich. Fast algorithm for n-2 contingency problem. In *2013 46th Hawaii International Conference on System Sciences*, pages 2161–2166, 2013.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[52] R. Vykuka and L. Nohácová. Sensitivity factors for contingency analysis. In *2015 16th International Scientific Conference on Electric Power Engineering (EPE)*, pages 551–554, 2015.

[53] Duotong Yang. A power system network partition framework for data-driven regional voltage control. In *2017 North American Power Symposium (NAPS)*, pages 1–6, 2017.

[54] Yuxuan Yuan, Yifei Guo, Kaveh Dehghanpour, Zhaoyu Wang, and Yanchao Wang. Learning-based real-time event identification using rich real pmu data, 2020.

[55] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks, 2019.

[56] J. Y. Zhang and C. M. Bush. Pmu based islanding detection to improve system operation. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, pages 1–5, 2016.

[57] Qianzhi Zhang, Kaveh Dehghanpour, Zhaoyu Wang, and Qiuhua Huang. A learning-based power management method for networked microgrids under incomplete information. *IEEE Transactions on Smart Grid*, 11(2):1193–1204, 2020.

[58] Yagang Zhang, Zengping Wang, and Jinfang Zhang. Pca fault location based on wide area measurement systems. In *2010 5th International Conference on Critical Infrastructure (CRIS)*, pages 1–4, 2010.

[59] Yiting Zhao, Chen Yuan, Guangyi Liu, and Ilya Grinberg. Graph-based preconditioning conjugate gradient algorithm for "n-1" contingency analysis. In *2018 IEEE Power Energy Society General Meeting (PESGM)*, pages 1–5, 2018.

[60] Xiangtian Zheng, Bin Wang, and Le Xie. Synthetic dynamic pmu data generation: A generative adversarial network approach. In *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, pages 1–6, 2019.

[61] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications, 2021.

[62] Mengze Zhou, Yuhui Wang, Anurag K. Srivastava, Yinghui Wu, and P. Banerjee. Ensemble-based algorithm for synchrophasor data anomaly detection. *IEEE Transactions on Smart Grid*, 10(3):2979–2988, 2019.

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