

Syracuse University

SURFACE at Syracuse University

Dissertations - ALL

SURFACE at Syracuse University

Spring 5-15-2022

Enhancing Grid Reliability With Phasor Measurement Units

Rui Ma

Syracuse University

Follow this and additional works at: <https://surface.syr.edu/etd>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Ma, Rui, "Enhancing Grid Reliability With Phasor Measurement Units" (2022). *Dissertations - ALL*. 1426.
<https://surface.syr.edu/etd/1426>

This Dissertation is brought to you for free and open access by the SURFACE at Syracuse University at SURFACE at Syracuse University. It has been accepted for inclusion in Dissertations - ALL by an authorized administrator of SURFACE at Syracuse University. For more information, please contact surface@syr.edu.

Abstract

Over the last decades, great efforts and investments have been made to increase the integration level of renewable energy resources in power grids. The New York State has set the goal to achieve 70% renewable generations by 2030, and realize carbon neutrality by 2040 eventually. However, the increased level of uncertainty brought about by renewables makes it more challenging to maintain stable and robust power grid operation. In addition to renewable energy resources, the ever-increasing number of electric vehicles and active loads have further increased the uncertainties in power systems. All these factors challenge the way the power grids are operated, and thus ask for new solutions to maintain stable and reliable grids. To meet the emerging requirements, advanced metering infrastructures are being integrated into power grids that transform traditional grids into "smart grids". One example is the widely deployed phasor measurement units (PMUs), which enable generating time-synchronized measurements with high sampling frequency, and pave a new path to realize real-time monitoring and control in power grids. However, the massive data generated by PMUs raises the questions of how to efficiently utilize the obtained measurements to understand and control the present system. Additionally, to meet the communication requirements between the advanced meters, the connectivity of the cyber layer has become more sophisticated, and thus is exposed to more cyber-attacks than before. Therefore, to enhance the grid reliability with PMUs, robust and efficient grid monitoring and control methods are required.

This dissertation focuses on three important aspects of improving grid reliability with PMUs: (1) power system event detection; (2) impact assessment regarding both steady-state and transient stability; and (3) impact mitigation. In this dissertation, a comprehensive introduction of PMUs in the wide-area monitoring system, and comparisons with the existing supervisory control and data acquisition (SCADA) systems are presented first. Next, a data-driven event detection method is developed for efficient event detection with

PMU measurements. A text mining approach is utilized to extract event oscillation patterns and determine event types. To ensure the integrity of the received data, the developed detection method is further designed to identify the fake events, and thus is robust against cyber-threat. Once a real event is detected, it is critical to promptly understand the consequences of the event in both steady and dynamic states. Sometimes, a single system event, e.g., a transmission line fault, may cause subsequent failures that lead to a cascading failure in the grid. In the worst case, these failures can result in large-scale blackouts. To assess the risk of an event in steady state, a probabilistic cascading failure model is developed. With the real-time phasor measurements, the failure probability of each system component at a specific operating condition can be predicted. In terms of the dynamic state, a failure of a system component may cause generators to lose synchronism, which will damage the power plant and lead to a blackout. To predict the transient stability after an event, a predictive online transient stability assessment (TSA) tool is developed in this dissertation. With only one sample of the PMU voltage measurements, the status of the transient stability can be predicted within cycles. In addition to the impact detection and assessment, it is also critical to identify proper mitigations to alleviate the failures. In this dissertation, a data-driven model predictive control strategy is developed. As a parameter-based system model is vulnerable to topology errors, a data-driven model is developed to mimic the grid behavior. Rather than utilizing the system parameters to construct the grid model, the data-driven model only leverages the received phasor measurements to determine proper corrective actions. Furthermore, to be robust against cyber-attacks, a check-point protocol, where past stored trustworthy data can be used to amend the attacked data, is utilized. The overall objective of this dissertation is to efficiently utilize advanced PMUs to detect, assess, and mitigate system failure, and help improve grid reliability.

Enhancing Grid Reliability with Phasor Measurement Units

By

Rui Ma

B.S., Central South University, 2014

M.S., The Ohio State University, 2015

Dissertation

Submitted in partial fulfillment of the requirements for the degree
of Doctor of Philosophy in Electrical & Computer Engineering

Syracuse University

May 2022

Copyright ©Rui Ma, 2022

All Rights Reserved

Acknowledgements

First, I would like to take this opportunity to express my deepest appreciation to my advisor Prof. Sara Eftekharnjad, for her support during my doctoral study. Her invaluable insight into research and the consistent encouragement have inspired me to overcome difficulties in research and life. Thanks for her supportive understanding and always being available to help me. The completion of my dissertation would not have been possible without her guidance.

I must also thank my mother Yuying Liu and my father Zhendong Ma for their unconditional love, care, and encouragement to help me get through all the tough times throughout my graduate study.

I would like to extend my deepest gratitude to Prof. Cliff Ian Davidson, Prof. Reza Zafarani, Prof. Fanxin Kong, Prof. Makan Fardad, and Prof. Mustafa Cenk Gursoy, for being my committee members. Their valuable comments and suggestions have helped me significantly improve this dissertation.

I also want to thank my labmates Sagnik Basumalik, Wolf Peter, Mirjavad Hashemi, Guangyuan Shi, Cheng Lyu, Chekai Chang for always giving me support. I am also grateful to my friends for providing sustained backing and being there for me. Thank you all for giving me these wonderful memories.

Table of Contents

Acknowledgements	v
List of Figures	xiv
List of Tables	xvi
1 Introduction	1
1.1 General Introduction	1
1.2 Motivation	2
1.3 Research Scopes and Objectives	3
1.4 Supervisory Control and Data Acquisition (SCADA) System	5
1.5 PMUs in the Wide Area Monitoring System	6
1.6 Contribution of Dissertation	7
1.6.1 Chapter 2	7
1.6.2 Chapter 3	9
1.6.3 Chapter 4	9
1.6.4 Chapter 5	10
1.7 Publications	11
2 Detection of Power System Events Considering Cyber-Attacks	13
2.1 Introduction	13
2.2 Background	14
2.2.1 Event Classification	14

2.2.2	Cyber-Attack	16
2.3	Two-Phase Scheme for Classifying System Events	16
2.4	PMU Event Data Processing and Feature Extraction	17
2.4.1	Overview of the Data Processing of the Event	17
2.4.2	Symbolic Aggregation approxXimation (SAX)	19
2.4.3	Multivariate Bag-of-Pattern (MBOP)	21
2.4.4	Term frequency and document frequency method	24
2.4.5	Feature Extraction for Power System Events	26
2.5	False Data Attack Model	28
2.5.1	Modelling of the False Data Attack	28
2.5.2	False Data Creation	29
2.6	Simulation Results	31
2.6.1	PMU placement	31
2.6.2	Event Data Generation	31
2.6.3	Model evaluation	35
2.7	Discussion	37
2.7.1	Impact of Model Parameters	37
2.7.2	Impact of the Coverage and Placement of PMUs	39
2.7.3	Robustness to Noisy Measurements	40
2.7.4	Performance with Untrained False data	41
2.7.5	Comparison with Conventional Event Classification Methods	42
2.8	Conclusions	44
3	Impact Prediction of Power System Events in Steady State	45
3.1	Introduction	45
3.2	Background	46
3.3	AC-based Cascading Failure Model	47
3.4	Dynamic Failure Model	49

3.4.1	Overview of the Dynamic Failure Model	49
3.4.2	Estimating the Dynamic Interactions	50
3.4.3	Predicting the Cascade Propagation	53
3.4.4	Cascade Data Sufficiency	57
3.5	Simulation Results	59
3.5.1	Cascade Data Preparation	60
3.5.2	Evaluation Metrics	63
3.5.3	Determining Optimal Failure Model Parameters	66
3.5.4	Performance Analysis	68
3.6	Discussion	73
3.7	Conclusion	75
4	Impact Prediction of Power System Events in Transient State	77
4.1	Introduction	77
4.2	Background	79
4.3	Hierarchical GAN-based Transient Stability Assessment	81
4.3.1	Transient Stability Assessment	82
4.3.2	The Generative Adversarial Network	85
4.3.3	Gated Recurrent Units for Feature Extraction from PMU Time Series Data	87
4.3.4	The Developed HGAN-based TSA Method	88
4.4	Simulation Results	93
4.4.1	PMU Data Generation	93
4.4.2	Performance Analysis of the HGAN-based TSA	94
4.4.3	Impact of PMU Measurement Variations	99
4.5	Discussions	102
4.6	Conclusions	105

5	Impact Mitigation of Power System Failures	107
5.1	Introduction	107
5.2	Background	108
5.3	Power Flow Model	109
5.3.1	Model-based Linear DC Power Flow Model	109
5.3.2	Parameter-based Power Flow Model	111
5.4	Recovery-based Model-driven MPC model	114
5.4.1	State Recovery	115
5.4.2	RMPC Model Construction	116
5.4.3	Improving RMPC Scheme with Checkpointing Protocol	119
5.5	Recovery-based Data-driven MPC Model	121
5.5.1	Overview of the MPC Controller	121
5.5.2	Construction of the Data-driven MPC Model	122
5.5.3	Reducing Computational Time	123
5.5.4	Improving Attack-resiliency	126
5.6	Simulation Results	127
5.6.1	Scenario Setup	128
5.6.2	Model Parameter Setup	129
5.6.3	RD-MPC against Wrong Topology	131
5.6.4	RD-MPC against False Data	134
5.7	Discussion	135
5.7.1	Robustness to Noise	135
5.7.2	Computation Time	135
5.7.3	Limitation and Improvement	136
5.8	Conclusion	137
6	Summary and and Future Work	139
6.1	Summary	139

6.2	Future Research Directions	141
6.2.1	Cascading Failure Analysis in both Steady and Dynamic States .	141
6.2.2	Distributed Data-driven MPC for Alleviating Thermal Overloads .	142
6.2.3	Transient Event Sequence Prediction	142
	Nomenclature	143
	References	146
	Vita	169

List of Figures

1.1	The three topics of this proposal	3
1.2	Illustration of the SCADA system in power grids	6
1.3	Illustration of improving grid resilience using PMUs in power grids	8
2.1	A two-phase data driven scheme for classifying power system events	17
2.2	Illustration of PMU data processing and feature extraction	18
2.3	Symbolic Aggregation approXimation ($\alpha = 4, \gamma = 10, \omega = 110$, length of data = 110)	21
2.4	Converting MBOP frequency matrix into a TF-DF weighted matrix	27
2.5	Architecture of the data-driven model in power systems	29
2.6	Voltage and current waveforms for three types of data spoofing attacks	30
2.7	Four zones for IEEE 118-bus system	32
2.8	Voltage and current waveforms in the vicinity of various event types	34
2.9	Average classification accuracy (in percent) of six events for different SAX parameters for IEEE 30-bus system	36
2.10	Confusion matrix for the developed classification method for the IEEE 30-bus system (accuracy = 99.7%, $\alpha = 3, \gamma = 4, \omega = 60$)	37
2.11	Confusion matrix for the developed classification method for the IEEE 118-bus system (accuracy = 97.2%, $\alpha = 3, \gamma = 4, \omega = 60$)	39
2.12	An attack strategy not considered during the training	42
2.13	Classification accuracy for both clean and noisy data	43
3.1	Example of a cascade propagation	48

3.2	Development of a dynamic failure model for predicting cascades	50
3.3	An example of a propagation tree	54
3.4	An illustration of the subtree of component 3 in generation 5	57
3.5	Diagram of IEEE 30-bus system that has 30 buses and 41 transmission lines. Red nodes represent generator buses and blue nodes represent load buses.	60
3.6	Diagram of Illinois 200-bus system that has 200 buses and 245 transmission lines. Red nodes represent generator buses and blue nodes represent load buses.	61
3.7	Variations of the Frobenius norm calculated for the dynamic interaction matrix for IEEE 30-bus system	62
3.8	Variations of the Frobenius norm calculated for the dynamic interaction matrix for Illinois 200-bus system	63
3.9	F_1 score of prediction accuracy of total failures in IEEE 30-bus and Illinois systems with different $\varepsilon \in [0, 1]$	67
3.10	F_1 score of prediction accuracy of total failures in IEEE 30-bus and Illinois 200-bus systems with different bins (from 1 to 100)	68
3.11	Average precision with top N number of generations	72
3.12	Potential failures in each generation of a cascade in IEEE 30-bus system.	74
3.13	Potential failures in each generation of a cascade in Illinois 200-bus system.	75
4.1	The trajectories of the generator rotor angles (19 generators) and voltage magnitudes (118 buses) in IEEE 118-bus system. A three-phase short circuit has occurred on line 102 at 0.167s and is cleared after 5 cycles (0.083 s).	78
4.2	Illustration of the unstable transient after a fault.	78
4.3	The structure of the developed HGAN-based TSA	82

4.4	85
4.5	The structure of GRU	89
4.6	The structure of GAN k in the HGAN-based TSA	91
4.7	The cross-entropy loss and the squared error for each of the three GANs in the HGAN-based model.	96
4.8	The confusion matrix of each GAN in the HGAN-based TSA model . . .	97
4.9	Comparison of response time between the baseline methods and HGAN- based TSA method. In total, 20 PMUs are used in the IEEE 118-bus sys- tem. To achieve the same accuracy as the HGAN-based method, other methods would use more samples of PMU measurements and thus require longer response time.	100
4.10	The TSA classification accuracy of each sub-GAN under different noise conditions. The testing data are used for evaluation.	101
4.11	104
4.11	105
5.1	The developed data-driven power flow model	112
5.2	Checkpointing protocol illustration in handling attack detection delays . .	120
5.3	The RMPC architecture for mitigating cascades following a CCPA	120
5.4	RD-MPC architecture for mitigating thermal overloads after a FDIA . . .	121
5.5	Example of a control action reduction for the IEEE 14-bus system. In order to regulate $line_{2-4}$, control actions are applied to buses 1, 3, 4 and 8. The parameters in (5.14) are $\rho_1 = 5$ and $\rho_2 = 0.1$	125
5.6	The Checkpointing protocol for addressing the attack detection delays . .	127
5.7	Performance comparison for mitigating an overload on line 22 in 30-bus system. Line 27 is incorrectly assumed outaged. A time step is 60s. . . .	133
5.8	Mitigating an overload on line 22 in 118-bus system with a wrong system model, that incorrectly assumes lines 15 and 20 are outaged.	133

5.9	Performance comparison of the data-driven MPC in mitigating an overload on line 54 of the 200-bus system. Line 38 is assumed to be out of service in the wrong model.	133
5.10	Performance comparison after an overload on line 22 in 118-bus system. A successful attack hides the overload. Mitigation provided in Table 5.2. . .	134
5.11	Evaluation of the robustness of the data-driven MPC against noisy data in mitigating the overload on line 22 in 118-bus system. Four different SNR levels are considered.	135

List of Tables

1.1	Comparison between SCADA and PMU meters	7
2.1	Lookup table containing the breakpoints that divide a Gaussian distribution into 3 to 5 equal probable regions	20
2.2	BOW representation of time series shown in Fig. 2.3 (length of data = 110, $\alpha = 3, \gamma = 2, \omega = 60$)	23
2.3	MBOP representation of multivariate time series ($\alpha = 3, \gamma = 2$)	24
2.4	PMU placement of IEEE 30-bus system and IEEE 118-bus system	31
2.5	Number of studied events	34
2.6	Classification accuracy under different coverage and different placement of PMUs ($\alpha = 3, \gamma = 4, \omega = 60$)	40
3.1	Performance analysis of the proposed dynamic interaction model for the IEEE 30-bus system	70
3.2	Performance analysis of the proposed dynamic interaction model for the Illinois 200-bus system	71
3.3	Example of two real sequence of failures in two test systems	73
4.1	PMU placement in the IEEE 118-bus system	94
4.2	Parameters of the HGAN-based TSA model	95
4.3	TSA classification accuracy (F1 score) of each GAN in the HGAN-based TSA	97

4.4	TSA classification accuracy of the baseline methods and the HGAN-based TSA with only one sample of PMU measurements (20 PMUs are used) . . .	99
4.5	TSA classification accuracy (F1 score) of each GRU in the stacked GRU model	99
4.6	PMU location scenarios for investigating the impact of PMU placement on the developed TSA approach	101
4.7	The TSA classification accuracy of each GAN with different PMU locations. The testing data are used for evaluation. A total number of 16 PMUs are used	102
4.8	TSA classification accuracy of each sub-GAN with different number of PMUs	102
5.1	Details of the coordinated attack scenarios	129
5.2	Thermal overload mitigation scenarios, the optimal number of control actions, and the selected lines for three systems	131
5.3	Computation time of the data-driven MPC for Illinois 200-bus system. The mitigation scenario is given in Table 5.2.	136

Chapter 1

Introduction

1.1 General Introduction

As the world is stepping into the fourth industrial revolution, i.e., a cyber-physical system, electricity has become indispensable for human livelihood and manufacturing. For electrical power engineers, ensuring a reliable supply of electricity is always the first task. However, with the ever-increasing deployment of renewables and the expansion of the power grids, maintaining a reliable operation of the power grids becomes challenging with existing grid monitoring and control strategies. If a system failure cannot be detected and mitigated promptly, severe subsequent failures may occur and, in the worst case, lead to a large blackout and cause substantial social and economic loss. In addition to the physical aspect of the grid, as the power grids evolve to a cyber-physical system, the threat of cyber attacks has also increased, making the grids more vulnerable. For example, a malicious false data injection attack can bypass existing bad data detection filters, generate false events, lead to incorrect control actions, and jeopardize the reliability of the system [1, 2]. Facing these new challenges, it is now required to develop new tools to efficiently **detect**, **assess**, and **mitigate** system failures in the presence of cyber threats and thus improve the reliability of the grid.

1.2 Motivation

Electricity is one of the most critical elements of our daily life. As society is chasing efficiency in every aspect of life, continuously supplying stable electricity to consumers becomes more important. However, the rapid growth of the deployed renewables and electric vehicles, the enlarged diversity of energy consumption, and the expansion of the power grids have extremely stressed grid operation, and challenged the existing power grids. If the system operating conditions are not properly monitored and controlled, severe grid failures can occur, which can lead to large blackouts. An example of blackout due to loss of situational awareness is the Northeast blackout of 2003, which affected both America and Canada for seven hours [3]. A recent large blackout in Texas in 2021 has also shown that the uncertainties introduced by wind power can lead to a large power outage and cause \$200 billion economic loss [4]. Hence, real-time situational awareness is essential for a reliable power grid. Benefiting from the advances in the smart grid, it is now possible to efficiently monitor and control power systems. One of these advances is the deployment of Phasor Measurement Units (PMUs), which can provide system measurements, e.g. voltage and current phasors, with a sampling frequency of 120 frames per second. Given the real-time PMU data, system operators can better understand the system conditions and maintain a stable system.

Unfortunately, these PMU measurements are vulnerable to cyber-attacks. For example, a man-in-the-middle attack can tamper with the PMUs, bypass bad data detection, create fake system events, and lead to costly or even harmful control actions. For example, in Ukraine's grid failure event, malware is used to control substation breakers and inject false data through different communication protocols such as IEC 61850 and IEC 60870-5-101 [5, 6]. Therefore, it is imperative to enhance real-time situational awareness of the power grid, which motivates us to develop robust tools to detect, assess, and mitigate grid failures under the threat of cyber attacks, and improve grid reliability.

1.3 Research Scopes and Objectives

The three main topics of this dissertation are illustrated in Fig. 1.1. Given physical events, for example, transmission line failures, and possible cyber threats, this work develops robust tools to: 1) **identify** the types of events and determine the integrity of the data; 2) **assess** the impact of the event in terms of both steady and dynamic states; 3) find optimal **mitigations** to bring the grid back to normal operation.

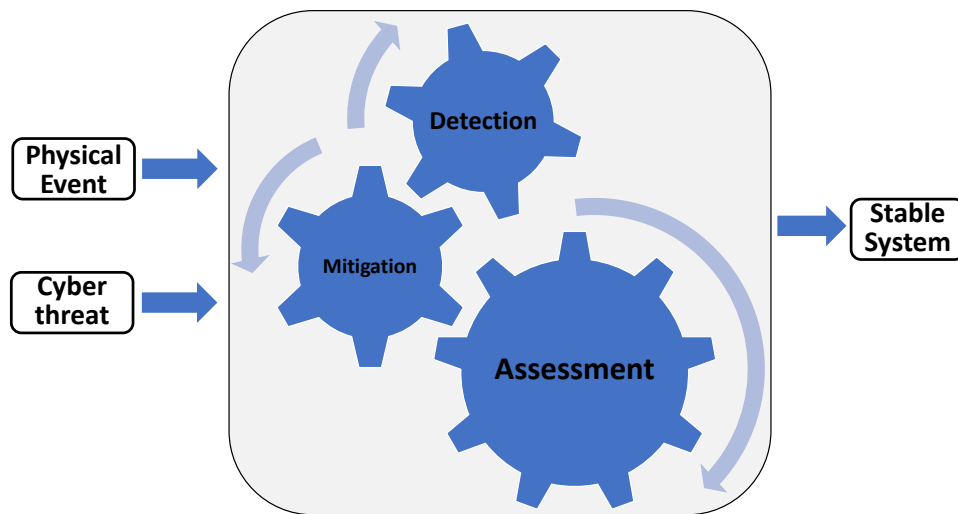


Figure 1.1: The three topics of this proposal

The objectives of these three topics are described as follows:

1. Event Detection

- When an event occurs, the event detector should be able to rapidly and accurately inform system operators of the types of system events;
- If the event is created by attackers, the event detector can efficiently distinguish the fake events from the real events;
- To increase the applicability in different grid sizes, the developed detection tool should be scalable and independent of the power network size;

2. Failure assessment

- Steady State:
 - Given the real-time PMU measurements, a failure model is developed to accurately and promptly predict failure propagation, i.e., sequence of the cascading failure;
 - As the propagation of cascades varies with different grid operating conditions, to provide more accurate failure information, the developed failure model should be adaptive to the changing operating conditions of the grids;
 - The failure model can provide the failure probability of each system component, and help grid operators identify critical components;
- Transient State:
 - The transient stability assessment tool can be used to accurately and promptly predict the transient status after the occurrence of a system failure;
 - By capturing the spatial and temporal correlations of the post-fault PMU time series data using machine learning methods, the transient stability assessment accuracy can be further improved.

3. Failure mitigation

- As the grid model information is threatened by cyber-attacks, a data-driven power flow model is constructed with only past PMU measurements;
- To alleviate system failures, in particular, transmission line overloads, a data-driven model predictive method is developed;
- To be robust to the false data attack, a check-point protocol is implemented to

correct tampered data with stored past PMU measurements.

1.4 Supervisory Control and Data Acquisition (SCADA) System

Conventionally, the power grids are monitored and controlled by supervisory control and data acquisition (SCADA) systems. SCADA systems are defined as a collection of equipment that allow system operators to remotely acquire accurate system states and control various components [7]. The SCADA in power grid has two main functions: (1) data acquisition of the grid operating condition; (2) supervisory control of the system components, e.g., switches. In summary, the SCADA system has four main elements, which are shown in Fig. 1.2 [7]:

- **Remote Terminal Unit (RTU)/ Data concentrator:** RTU or data concentrator is the link between the field and control center, where field data can be measured and sent to the control center, and command data can be passed from the control station to the field equipment;
- **Communication System:** The communication system is responsible for sending the acquired data from the RTU / Data concentrator to the control center and transmitting the command from the master station to the RTU/Data concentrator;
- **Master Station:** The master station is the place where operators can determine control actions and monitor the grid states;
- **Human Machine Interface (HMI):** HMI is the user interface that allows the interaction between the grid operators and the master station.

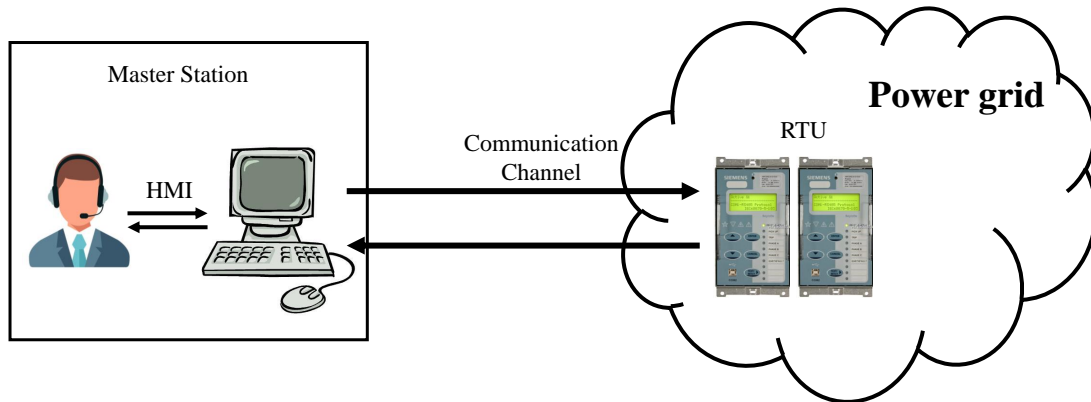


Figure 1.2: Illustration of the SCADA system in power grids

1.5 PMUs in the Wide Area Monitoring System

As mentioned in the previous section, SCADA system uses the sensors to observe various types of data, e.g., the magnitude of the bus voltage and line current, and send the measurements through the RTU to the master station. However, these measurements are not synchronized and the refreshing rate is relatively low (updated every 2 to 4 messages per second). With the increased deployment of PMUs at the current transformer (CT) and potential transformer (PT), the SCADA system is now shifting towards a wide-area monitoring system (WAMS). PMUs are electronic meters that can measure bus voltage and line current phasors, frequency, as well as the rate of change of frequency (ROCOF). Unlike SCADA meters, PMUs provide time-synchronized phasor measurements, with a sampling frequency that can range from 30 to 240 messages per second. Comparisons between SCADA and PMU are given in Table 1.1.

Table 1.1: Comparison between SCADA and PMU meters

	Measurements	Sampling Frequency (messages per second)	Synchronized by GPS
SCADA	Magnitude of the bus voltage, Magnitude of line current	2 to 4 (only steady state)	N/A
PMU	Phasors of the bus voltage and line current, frequency, rate of change of frequency (ROCOF)	30 to 240 (both steady and dynamic states)	Yes

An overview of the WAMS architecture is illustrated in Fig. 1.3. PMUs, deployed in CT and PT, continuously measure the voltage and current phasors, frequency, and ROCOF. Meanwhile, attackers can use malware to alter the measurements and send wrong information to the control center. According to the IEEE C37.118.2 standard, PMUs transmit GPS time-stamped phasor data to their local phasor data concentrator (PDC), where the data will be processed into streams. The output stream will be further sent to the super PDC, and used by the control center for various applications. Here, we focus on the applications of detection, assessment, and mitigation of the event, which belong to the two main functions of the WAMS, i.e., event analysis and wide area monitoring and control.

1.6 Contribution of Dissertation

The four chapters of this dissertation discuss the methods developed for event detection, assessment, and mitigation. The contributions of this research are explained below.

1.6.1 Chapter 2

Dynamic visualization is essential for power system operation and control, as it enables providing the cause of the events and facilitating the post-event decision-making process to prevent potential cascading failures. This chapter develops an effective method for power

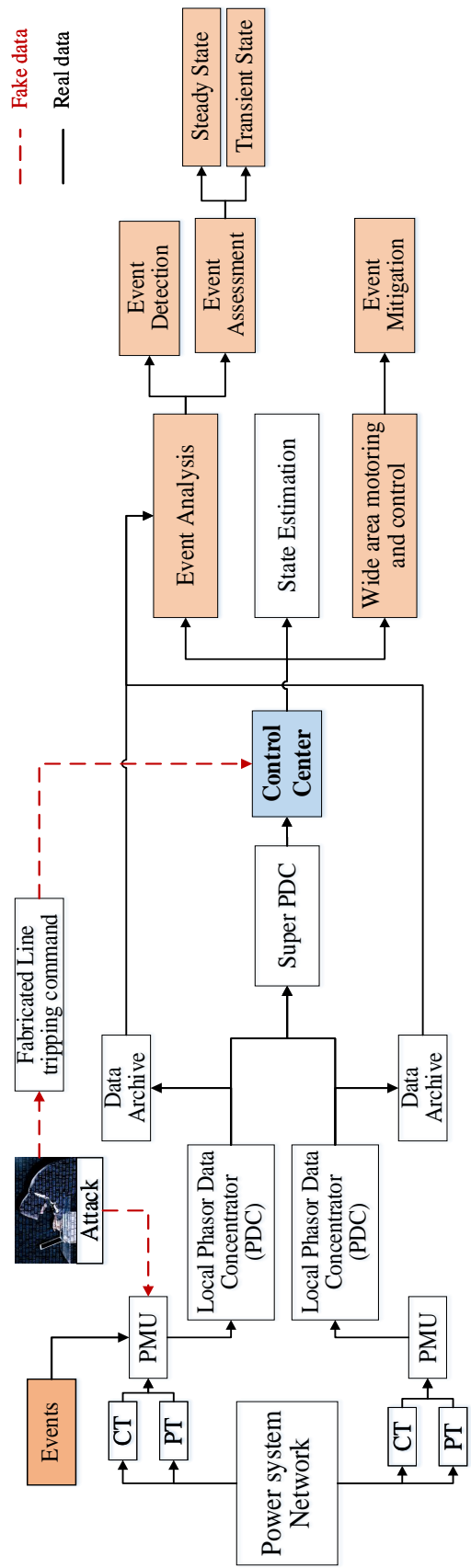


Figure 1.3: Illustration of improving grid resilience using PMUs in power grids

system event identification purely based on the received multivariate PMU time series data. For each power system event, a data processing method, i.e., Symbolic Aggregation approxiMation (SAX) method, is used to compress and convert PMU time-series data into text words that in turn are fed into the machine learning classifier to classify the types of events. In comparison to the conventional event classification methods, this methodology achieves a higher classification accuracy and has the ability to identify fake transients that are produced as a result of cyber-attacks. Furthermore, by using the substation measurement available at each local PDC, the developed event detection approach is scalable and independent of the size of the power network.

1.6.2 Chapter 3

When a system event is detected, it is also critical to estimate the consequences of the event and assess its impact. A small failure in the system may lead to a severe cascading failure. Therefore, to predict the potential cascading failure sequence under specific operating conditions, in this chapter, we develop a *dynamic cascading failure model* that is adaptive to changing operating conditions of the system. Bayesian framework is used to estimate the interactions of the components of the power grid failures for *a specific operating condition*, resulting in a more accurate estimation of the interactions for near real-time operations. Additionally, with the developed propagation method, the failure probability of each system component can be predicted at each point in time, and thus operators can help find risky components that need further protection. Compared with the DC/AC-based cascading failure model, this dynamic interaction model can forecast impending failures in a significantly shorter time.

1.6.3 Chapter 4

The aforementioned event assessment only looks at the steady-state conditions of the power grids. However, post-event transient status also plays a key role in maintaining

a stable grid, as unstable transients can lead to severe cascading failures and result in a large economic loss. In this chapter, we develop a new data-driven transient stability assessment (TSA) tool to efficiently predict the post-fault transient status. Specifically, the generative and adversarial networks (GAN) are used to learn the temporal correlations of the received PMU voltage time series data to make more accurate TSA decisions. With only one sample of the post-contingency PMU voltage data, the developed TSA can predict the evolution of the time series data, and thus determine the status of the transient at the earliest stages after a fault.

1.6.4 Chapter 5

If for any reason a failure in the power grids causes further thermal overloads on transmission lines, it is imperative to find optimal mitigations to alleviate overloads and prevent the propagation of cascading failures. However, it is now more challenging than ever to make prompt and accurate mitigations, as false data attacks on the measured data, or relay status information, may lead to incorrect estimation of the system operating condition. To address these new challenges, in Chapter 5, a data-driven model-predictive control (MPC) method is developed to effectively alleviate thermal overloads under false data injection. Here, we introduce a new measurement-based power flow solution to replace the approximate model-based DC power flow. In contrast to parametric data-driven methods that require calculating system parameters (such as Injection Shift Factor (ISF)), the developed data-driven method is non-parametric and thus is immune to wrong system topology information. By leveraging the concept of check point protocol, if a data attack is detected, the actual system states can be recovered from the historical trustworthy data. Hence, a more robust mitigation method is achieved. Furthermore, to make the control methods more suitable for online applications, two strategies are developed to simplify the control action space while retaining feasible solutions.

1.7 Publications

The research work presented in this dissertation has appeared in multiple publications, as outlined below:

Journals:

1. **Ma, Rui**, Sagnik Basumallik, and Sara Eftekharnjad. "A pmu-based data-driven approach for classifying power system events considering cyberattacks." *IEEE Systems Journal* 14.3 (2020): 3558-3569.
2. **Ma, Rui**, et al. "A Probabilistic Cascading Failure Model for Dynamic Operating Conditions." *IEEE Access* 8 (2020): 61741-61753.
3. **Ma, Rui**, et al. "A Data-Driven Model Predictive Control for Alleviating Thermal Overloads in the Presence of Possible False Data." *IEEE Transactions on Industry Applications* 57.2 (2021): 1872-1881.
4. **Ma, Rui**, et al. "A Predictive Online Transient Stability Assessment with Hierarchical Generative Adversarial Networks." *arXiv preprint arXiv:2112.07840* (2022).

Conferences:

1. **Ma, Rui**, and Sara Eftekharnjad. "Critical PMU measurement identification based on analytic hierarchy process." 2018 IEEE/PES Transmission and Distribution Conference and Exposition (TD). IEEE, 2018.
2. **Ma, Rui**, and Sara Eftekharnjad. "A PMU-based Fault Location Identification Using Principal Component Analysis." 2019 IEEE Power Energy Society General Meeting (PESGM). IEEE, 2019.
3. **Ma, Rui**, et al. "Recovery-based Model Predictive Control for Cascade Mitigation un-

der Cyber-Physical Attacks.” 2020 IEEE Texas Power and Energy Conference (TPEC). IEEE, 2020.

4. **Ma, Rui**, et al. ”A PMU-based Data-Driven Approach for Estimating the Injection Shift Factors.” 2021 IEEE Texas Power and Energy Conference (TPEC). IEEE, 2021.
5. **Ma, Rui**, and Sara Eftekhariad. ”Data Generation for Rare Transient Events: A Generative Adversarial Network Approach.” 2021 IAS Annual Meeting. IEEE, 2021.

Chapter 2

Detection of Power System Events

Considering Cyber-Attacks

2.1 Introduction

This chapter concerns the identification of the types of transient events in the presence of cyber threats. The operation and control of the power system have become more complex with the increased levels of uncertainty associated with renewable energy resources and the growing interconnectivity of the grids. Dynamic visualization to detect and identify events in the power system is a step toward addressing the increase in system complexity for real-time operation and control [8]. Benefiting from the high sampling frequency of the PMUs, the transient events can now be captured and visualized. However, there also comes a concern of cyber-attack in PMU measurements. A malicious attack can bypass existing bad data detection algorithms and create fake transients that prompt system operators to take unnecessary or even harmful control actions. Hence, it is necessary to have a fast, accurate, and reliable power system event classification tool to recognize these types of events, and at the same time distinguish the *fake* events from the *real* events, such that no additional verification is needed by the operators.

This chapter introduces a methodology for power system event identification purely based on data received from PMUs, where the data are processed and analyzed to identify the types of transient events that have occurred within the system. For each power system event, a multivariate time series data is extracted from temporal PMU measurements *ag-*

gregated at the phasor data concentrator (PDC) to analyze time series discords, i.e., the event oscillation patterns. Due to the promising performance in detecting time series discords and data pruning ability, the Symbolic Aggregation approXimation (SAX) method is used to compress and convert PMU time series data into representations that are suitable for feature extraction and pattern identification. These features and oscillation patterns are found for each transient event, namely false data attacks, faults, line tripping, load change, generation change, and shunt switching. Using the extracted features, machine learning mechanisms, namely the Support Vector Machine (SVM) and the Ensemble classifier based on Bagged Trees, are used to classify these six types of events.

2.2 Background

2.2.1 Event Classification

Researchers have investigated the use of PMU data to identify events in the power systems, such as faults, line tripping, and generation loss [8–16]. Depending on how the data is processed, the developed methods can be grouped into three categories: signal processing methods, heuristic methods, and time-series analysis methods. Traditional signal processing techniques such as wavelet analysis, short time Fourier transform (STFT) and Hilbert analysis are utilized to analyze the oscillation modes of PMU signals under different transient events [9–11]. However, signal processing methods require an appropriate sampling rate and a proper length for the sample window to guarantee an acceptable event detection performance [15]. As an alternative, a heuristic event identification method is developed in [12] and [13] where all PMU data are enclosed into an ellipsoid, so that features such as volume and orientation can be used to identify power system disturbances. Heuristic methods, on the other hand, require significant classification time, challenging their application for real-time operations [14]. Instead of using all PMU data accumulated in PDC, authors in [8] and [16] develop a time-series analysis method that utilizes data

from a single PMU with the largest variation to promptly identify the type of transient events. Using the time series sub-sequences called shapelets that maximally represent PMU data [17], this time-series approach identifies disturbance patterns while enhancing the classification accuracy. Although the aforementioned event classification techniques are promising in terms of accuracy, the challenge of identifying *fake* events due to false data attacks has not been addressed to the best of our knowledge. Hence, the credibility of event classification methods might become questionable under certain circumstances. This particular challenge is addressed.

The availability of PMU-generated time series data creates an opportunity to deploy classification techniques to identify false data, as evidenced by many research efforts. Most classification techniques can be divided into two categories: shape-based methods and structural-based methods [18]. Shape-based methods use the distance between two time-series data points to detect similarities (or differences). Euclidean distance and Dynamic Time Warping (DTW) are the two common shape-based techniques used to quantify similarity [19]. These methods perform well in short time series, while the results in long time series are less promising, or require a long computation time [20]. On the contrary, structural-based methods use global features, such as sparsity or entropy, to compare similarity between long time series instead of analyzing each point of time-series data. A widely used structural-based method, Bag-of-Pattern (BOP), uses the sparsity of each time series to measure similarity [21]. Nevertheless, the aforementioned shape-based and structural-based classification methods focus on a univariate classification problem where the correlation among time-series data is not considered. For classifying power system events including fake transients, analyzing the correlation between different PMU time series data is imperative since a false data attack on PMUs will cause the data to lose correlation [22].

2.2.2 Cyber-Attack

Cyber-attacks on power systems can be conducted for various reasons. For example, measurements can be altered or circuit breakers disabled to disrupt power grid operation or to acquire economic benefits [23]. The authors in [24] have shown that malware is capable of controlling substation breakers and injecting false data through different communication protocols such as IEC 61850 and IEC 60870-5-101. A GPS spoofing attack on PMUs is believed to be one of the most threatening cyber-attacks on smart grids where the synchronization of measurements is disrupted [25]. Authors in [26–29] show that a malicious cyber-attack on PMUs can remain undetected and authors in [30], [22] and [31] have shown that attackers can mimic a transient event to inject false data into a set of PMU streams, and thus cause wrong operator actions due to incorrect event information.

2.3 Two-Phase Scheme for Classifying System Events

In this event detection study, a two-phase, data-driven event classification scheme is developed to enable fast online classification of power system transient events, as illustrated in Fig. 2.1. In the first phase, which is carried out offline, simulated multivariate PMU time series data under different system events train an event classifier. Specifically, features such as the oscillation patterns observed in the event data are first extracted from the multivariate data. These features are then used by machine learning classifiers to learn the patterns specific to each type of event. The second phase, which is carried out online, is triggered when the system signals such as voltage and frequency are beyond the pre-defined event detection threshold. During this phase, local PDCs will record the event data from PMUs, where the time-series data are analyzed to extract features that are specific to that event. An example of these features are the signal oscillation patterns and the correlation between PMU data. Once the features are extracted, the event classifier trained in the first phase can be used to determine the type of the detected event. Eventually, this

scheme helps power system operators better understand the cause of the events once they occur.

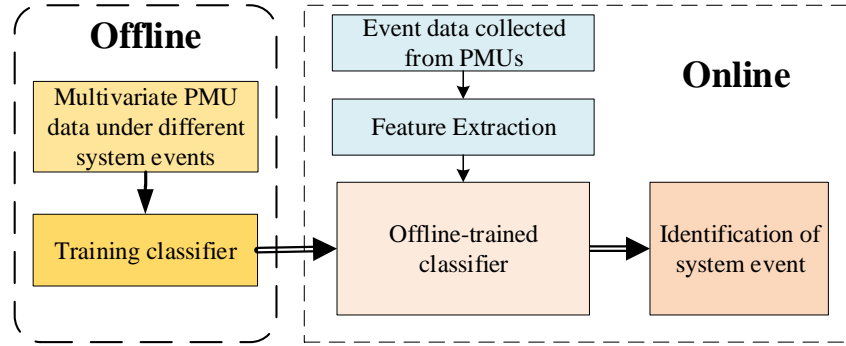


Figure 2.1: A two-phase data driven scheme for classifying power system events

2.4 PMU Event Data Processing and Feature Extraction

2.4.1 Overview of the Data Processing of the Event

In this section, a time series discord discovery method is developed to process PMU data and uncover patterns specific to each event. Specifically, PMU time series data are pruned and converted into SAX words, which in turn are used to create a multivariate BOP (MBOP) matrix. Finally, a method, referred to as term frequency and document frequency (TF-DF) method, is developed to highlight the oscillation patterns, and quantify the correlation between PMU measurements, using the constructed MBOP matrix. Thus, event features can be extracted *rapidly* with the developed methodology.

PMU data processing and the feature extraction process are illustrated in Fig. 2.2. By applying SAX to the labeled multivariate PMU voltage time series after defining the three SAX parameters α , γ , and ω , PMU data is converted to a set of symbolic words. Next, a multivariate BOP matrix M_{BOP} is constructed where the number of the columns equals to the number of PMU voltage time series and the size of each column is α^γ . The element $M_{BOP_{i,j}}$ denotes the number of times a SAX word in row i occurs in j th PMU voltage time

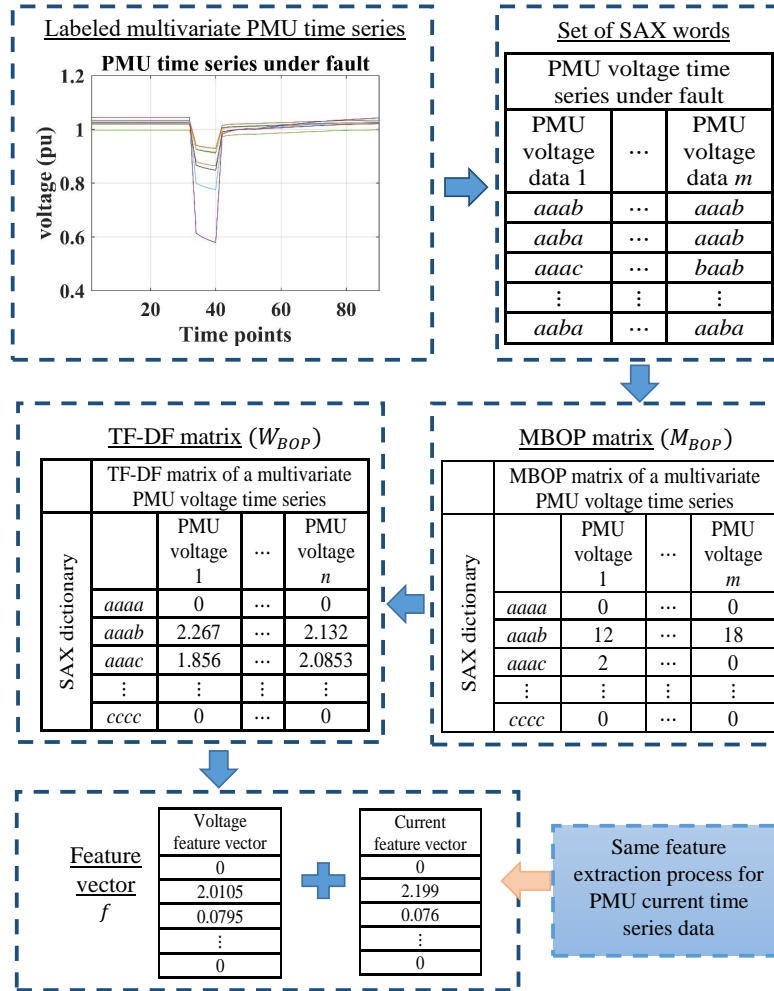


Figure 2.2: Illustration of PMU data processing and feature extraction

series. Therefore, the oscillation patterns of the PMU voltage time series are converted to frequencies and stored in the M_{BOP} matrix. After forming M_{BOP} , the modified TF-DF method converts the frequency of each SAX word to a weight. As a result, a weighted matrix W_{BOP} is constructed and converted to a voltage feature vector. Following the same process as the multivariate PMU voltage data, a current feature vector can be computed and combined with the voltage feature vector to construct the event feature vector f eventually.

2.4.2 Symbolic Aggregation approXimation (SAX)

As PMU measurements are updated with high frequency, massive data will be generated. For example, authors in [32] point out that 100 PMUs can generate up to 50 GB of data per day. This volume of data poses major computational and hardware challenges for analyzing the recorded PMU data. Due to the superior performance in dimension reduction, computation time, and lower bound ability [21], SAX is used in this study to compress PMU data, while guaranteeing the correctness of the similarity search in the reduced space. This data reduction method enables pruning PMU data into symbolic words such that, with proper text mining algorithms, time series discords can be rapidly identified. Next, the fundamentals of the SAX method are described.

Given a PMU time series data T of length n and a sliding window of length ω ($\omega \leq n$), SAX converts T into $(n - \omega + 1)$ number of overlapping sub-sequences by continuously moving the window a unit distance ahead from the initial point of T . For each sub-sequence data p , SAX first normalizes p such that its mean is zero and its standard deviation is one. This property creates a distinct advantage for SAX over other methods that suffer from variance scale issues for event classification problems. For instance, given a small and a large generation trip that have similar waveforms but different variance scales, this normalization strategy enables minimizing the difference between these two variance scales such that the impact of variance scales on pattern recognition is minimized. Next, the normalized sub-sequence p is divided into γ number of equally-sized segments, where the mean value of each segment is recorded. Hence, instead of using ω number of data points to represent the sub-sequence p , p can be denoted as a set of γ number of mean values, where γ is defined as the size of a word. The process of aggregating each mean value to represent the sub-sequence p , known as Piecewise Approximate Aggregations (PAA), significantly reduces the size of the data [33]. After forming a PAA representation of p , the mean value of each segment is converted into an alphabet letter ($a, b, c, etc.$) by setting

the alphabet size α and comparing the mean with a set of breakpoints that are presented in Table 2.1 [34]. All mean values of the PAA representation that are less than the smallest breakpoint are denoted as a , mean values that are larger or equal than the smallest breakpoint but less than the second smallest breakpoint are represented as b , etc. [34]. Therefore, a symbolic word is constructed to represent the sub-sequence p . For instance, given $\alpha = 3$ and the PAA representation of p as $[0.5, 0.2, -0.5]$, the sub-sequence p can be represented as a symbolic word cba . After transforming each sub-sequence into a symbolic word, a time series T is converted into $(n - \omega + 1)$ number of words.

Table 2.1: Lookup table containing the breakpoints that divide a Gaussian distribution into 3 to 5 equal probable regions

		Alphabet Size (α)		
		3	4	5
Breakpoint	β_1	-0.43	-0.67	-0.84
	β_2	0.43	0	-0.25
	β_3		0.67	0.25
	β_4			0.84

Three parameters are required to convert time series into symbols: the alphabet size (α), the size of a word produced by SAX (γ), and the length of the sliding window (ω). The size of the alphabet is the number of different alphabet letters that are used to represent a time series. The word size is the number of letters present in the output from the sliding window. The length of the sliding window is the number of the points of a time series that are to be converted. Given these three parameters and following the breakpoints presented in Table 2.1, a set of words is created as described in the aforementioned steps to represent a time series. For example, given a time series of length 110 in Fig. 2.3 after normalization, one can divide this time series data into 11 equally-sized segments by setting the size of a word γ as 10 and the sliding window ω as 110. Then, given the alphabet size $\alpha = 4$, the mean value of each segment is calculated and converted to a letter by comparing the

mean with three breakpoints $\beta_1 = -0.67$, $\beta_2 = 0$ and $\beta_3 = 0.67$. Thus, this time series data can be converted into a symbolic word *bcbabddcaa*, reducing the data size from 110 to 10. If the size of the sliding window is changed to 109, two SAX words, *bcbabdddaa* and *bcbabddcaa*, can be generated to represent these time series data.

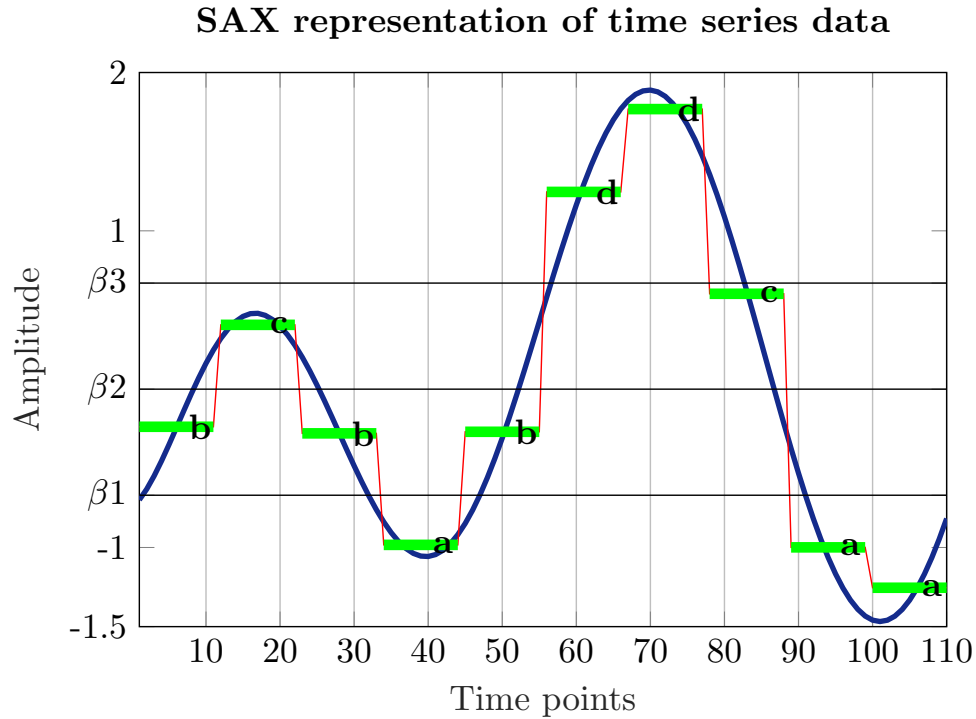


Figure 2.3: Symbolic Aggregation approXimation ($\alpha = 4, \gamma = 10, \omega = 110$, length of data = 110)

2.4.3 Multivariate Bag-of-Pattern (MBOP)

Once PMU time series data is compressed and converted into SAX words, the occurrence of each SAX word is used to find time series discords, i.e., transient event oscillation patterns. For this purpose, a methodology referred to as MBOP, is developed. MBOP is based on a text mining technique, i.e., Bag-of-Words (BOW) that creates feature vectors from the generated SAX words.

Conventional text mining techniques such as BOW method enable converting SAX words generated from a uni-variate PMU time series into a frequency vector, by counting the

occurrence of each SAX word. Thereby, the oscillation patterns of univariate time-series data can be represented as a frequency vector. BOW, inspired by information retrieval and text mining, has been applied to uni-variate time series classification in many prior works [18, 34, 35].

Given the size of the alphabet α , the size of the word γ and the length of the sliding window ω , one can convert a uni-variate PMU time series T of length n into $(n - \omega + 1)$ number of SAX words. Then a word-sequence frequency column vector of length α^γ , where each element of this vector corresponds to a specific SAX word, can be constructed to represent the univariate time series data. For instance, given $\alpha = 2$ and $\gamma = 2$, the size of the frequency vector is $2^2 = 4$ and the elements from the first to the fourth row of this vector correspond to SAX words aa , ab , ba , and bb , respectively. Each element in this frequency vector represents the number of occurrences of the corresponding word among all the $(n - \omega + 1)$ generated SAX words. The word-sequence frequency vector is the BOW representation of a univariate PMU time series T . The time series in Fig. 2.3 can be converted into 51 SAX words by setting ω to 60. The size of the alphabet is equal to three and the size of the word is equal to two, therefore, the size of the column vector is $3^2 = 9$. By counting the number of times each word occurs (e.g., the word ac occurs 23 times) among the 51 SAX words, the BOW vector, shown in Table 2.2, can be constructed as a feature vector for this time series data.

Although BOW representation can help create feature vectors from the generated SAX words, this method is not adequate for finding oscillation patterns of multivariate data as the correlations between variables are not considered. To identify false data, *the correlation between synchronized PMU data becomes a key identifying factor*. During a transient event, all PMU measurements exhibit strong correlation. On the contrary, in a fake transient, the correlation between measurements drops since false data attacks can only change a subset of PMU measurements. Thus, it is necessary to develop a method that can

Table 2.2: BOW representation of time series shown in Fig. 2.3 (length of data = 110, $\alpha = 3, \gamma = 2, \omega = 60$)

SAX dictionary	Frequency
<i>aa</i>	0
<i>ab</i>	0
<i>ac</i>	23
<i>ba</i>	0
<i>bb</i>	20
<i>bc</i>	0
<i>ca</i>	8
<i>cb</i>	0
<i>cc</i>	0

discover the event patterns hidden in each individual PMU data while considering the correlation among these data. A BOP representation based on BOW has been defined in [36] to classify multivariate physiological data by capturing the correlation of multivariate sensor time series. However, the features of each individual time series data are ignored when constructing the BOP representation.

Contrary to the BOP representation, we develop the MBOP representation where the information from each PMU time series data and the correlation between them are retained. A MBOP matrix $M_{BOP} = [BOW_{PMU_1}, BOW_{PMU_2}, \dots, BOW_{PMU_n}]$, which is a combination of several BOWs, as illustrated in Table 2.3, is created to represent multivariate PMU time series data. Each row of M_{BOP} denotes a SAX word and each column corresponds to a PMU time series. The elements of M_{BOP} represent the number of times a SAX word occurs in the corresponding PMU data. Intuitively, the oscillation patterns observed in the measured signals after an event are quantified as the frequencies of SAX words in M_{BOP} .

Table 2.3: MBOP representation of multivariate time series ($\alpha = 3, \gamma = 2$)

		Multivariate PMU time series				
		PMU1	PMU2	PMU3	...	PMU n
SAX dictionary	aa	0	0	0	...	1
	ab	20	18	16	...	17
	ac	15	16	17	...	18

	cc	0	0	2	...	0

2.4.4 Term frequency and document frequency method

The MBOP representation enables transferring multivariate PMU time series data into a frequency matrix using SAX such that the event features, e.g. oscillation patterns and correlation between PMUs, can be further extracted. However, the distinct patterns for each event and the data correlation are not readily available through the M_{BOP} matrix. In order to uncover these patterns that are hidden in the M_{BOP} matrix and quantify the correlation between the measurements, a modified TF-DF method is developed. Specifically, TF identifies the oscillation patterns of PMU data while DF quantifies the correlation between PMU data. Traditional term frequency and inverse document frequency (TF-IDF) method is a weighting approach for evaluating the significance of a word in a document, and is deployed in [18] to classify univariate time series in a way that words unique to a time series are highlighted. In the context of this chapter, for a SAX word t , TF quantifies the number of times t occurs in a PMU time series while IDF represents the inverse fraction of the number of PMUs containing the word t . Contrary to the TF-IDF method, the TF-DF method aims to recognize the hidden patterns for each event signals that most PMU measurements record as these measurements are highly correlated during a real transient event (except during a fake transient). For a fake transient event, as the PMU measurements are less correlated, the decreased degree of correlation among PMU data can be captured by

TF-DF technique to detect false data.

Term frequency and inverse document frequency are calculated as follows [18],

$$tf_{t,d} = \begin{cases} 1 + \log(f_{t,d}), & \text{if } f_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.1)$$

$$idf_{t,D} = \log \frac{N}{|\{d \in D : t \in d\}|} = \log \frac{N}{df_t} \quad (2.2)$$

where t is a SAX word extracted from a PMU time series, d is a PMU time series, $f_{t,d}$ is the number of times t occurs within the SAX words generated from the PMU time series d , D represents a set of PMU time series data, N is the total number of PMUs in D , and df_t is the number of PMU time series that contain the SAX word t . It can be observed in (2.1) and (2.2) that the weight $tf_{t,d}$ would be increased if the number of occurrences $f_{t,d}$ increases in the PMU time series d , and the weight $idf_{t,D}$ would be increased if the rarity of the SAX word t , i.e. $\frac{N}{df_t}$, in all PMU time series data D increases.

In the TF-IDF method, IDF assigns higher weights to unique words as those are more informative than frequent words. On the contrary, the event classification problem in this study assigns higher importance (weights) to the common words among PMU time series as these data are strongly correlated. Hence, a scaled document frequency (DF), counting the number of PMU time series data containing word t , is defined as,

$$df_{t,D} = \exp\left(\frac{|\{d \in D : t \in d\}|}{N} - 1\right) = \exp\left(\frac{df_t}{N} - 1\right) \quad (2.3)$$

It can be observed in (2.3) that the weight of t would be higher if more PMU time series data contain the SAX word t , and can be equal to one if t occurs in all PMU data.

The TF-DF weight of t in a PMU time series d is the product of two factors: term frequency

and document frequency,

$$tf \times df = (1 + \log(f_{t,d})) \times \exp\left(\frac{df_t}{N} - 1\right) \quad (2.4)$$


The weight of t with respect to TF-DF increases with the number of $f_{t,d}$ occurrences in PMU data d , as well as the number of PMUs df_t containing the SAX word t . Hence, the TF-DF method efficiently identifies oscillation patterns of a multivariate PMU data by assigning higher weights to the words that can represent most PMU time series. After transferring each PMU time series into a set of SAX words, TF counts the number of times a SAX word appears in the corresponding PMU time series. DF counts the number of PMU time series that contain a SAX word. It converts the frequency matrix obtained from MBOP into a weighted matrix such that the word that represents most of the PMUs within a multivariate PMU time series is assigned a higher weight, while a lower weight is assigned to a word unique to a single PMU data. Once the TF-DF weighted matrix is constructed, features can be extracted to facilitate the classification. Given a M_{BOP} matrix in Fig. 2.4, the TF-DF weighted matrix can be constructed based on (2.4). For example, the words ca and cb have the same number of occurrences in the first PMU time series. However, the TF-DF method assigns a higher weight to ca than cb due to the fact that ca occurs in data from both PMUs while cb only appears in the first PMU data.

2.4.5 Feature Extraction for Power System Events

Given a TF-DF matrix W_{BOP} obtained from a multivariate PMU voltage time series data, a column vector can be extracted from the weighted matrix W_{BOP} , where the element of the vector is the mean of the corresponding row of W_{BOP} , which is given as,

$$f_{W_{BOP},i} = \frac{\sum_{j=1}^n W_{BOP,i,j}}{n} \quad (2.5)$$

MBOP matrix of a PMU multivariate time series			
		PMU time series 1	PMU time series 2
		SAX dictionary	<i>aa</i>
<i>ab</i>	0		1
<i>ac</i>	0		1
<i>ba</i>	20		19
<i>bb</i>	17		18
<i>bc</i>	2		1
<i>ca</i>	1		2
<i>cb</i>	1		0
<i>cc</i>	0		0



TF-DF matrix of a PMU multivariate time series			
		PMU time series 1	PMU time series 2
		SAX dictionary	<i>aa</i>
<i>ab</i>	0		0.6065
<i>ac</i>	0		0.6065
<i>ba</i>	2.3010		2.2788
<i>bb</i>	2.2304		2.2553
<i>bc</i>	1.3010		1
<i>ca</i>	1		1.3010
<i>cb</i>	0.6065		0
<i>cc</i>	0		0

Figure 2.4: Converting MBOP frequency matrix into a TF-DF weighted matrix

where $f_{W_{BOP},i}$ is the i th element of the column vector, and n is the total number of the columns in the weighted matrix W_{BOP} . Authors in [34] point out that data from different clusters have different sparsity levels in the BOP matrix. Therefore, in addition to the mean vector $f_{W_{BOP}}$, the sparsity level of the SAX word in M_{BOP} matrix is also extracted as a feature. In this work, a cluster represents an event of the power system. Thus, the sparsity level vector f_s of the W_{BOP} matrix, which is given in (2.6), is combined with the mean vector $f_{W_{BOP}}$ to construct a voltage feature vector $f_{voltage}$.

$$f_{S_i} = \frac{\|W_{BOP_i}\|_0}{n} \quad (2.6)$$

$$f_{voltage,i} = f_{S_i} \times f_{W_{BOP},i} \quad (2.7)$$

where f_{S_i} is the i th element of the sparsity level vector, $\|W_{BOP_i}\|_0$ is the number of nonzeros in the i th row of the weighted matrix W_{BOP} , n is the total number of columns in the W_{BOP} and $f_{voltage,i}$ is the i th element of the voltage feature vector. The PMU current feature vector can be extracted from the labeled multivariate PMU current time series data following the same feature extraction process. Finally, the labeled *hybrid* multivariate

PMU time series feature vector, $f^T = [f_{current}^T, f_{voltage}^T]$, is constructed by combining the voltage feature vector $f_{voltage}$ and the current feature vector $f_{current}$.

2.5 False Data Attack Model

2.5.1 Modelling of the False Data Attack

Power systems are designed based on the well-established $N - 1$ criterion and are equipped with protection mechanisms to withstand and detect the aforementioned attacks. Bad data detection in energy management systems (EMS) is one example of such detection schemes that is intended to identify data anomalies that exist due to various reasons. In order to bypass a bad data detection scheme, an attacker has to launch a *coordinated attack* where multiple parts of a system have to be compromised in the cyber or physical layers. Various coordinated attack strategies have been developed in [37], such as coordinated physical attacks against lines and coordinated cyber attacks against substations. In this chapter, a coordinated attack strategy is adopted and illustrated in Fig. 2.5, in which both PMU measurements and the relay signal are tampered by the attackers. The aim of the attackers is to misinform the operators, to think an actual contingency has occurred in power systems, where in reality that contingency does not exist, hence cannot be detected by a protection relay. For instance, in order to successfully inject a coordinated attack to mimic a real scenario of a line fault, the compromised PMU time series data has to be similar to real power system line fault event data. A state estimator that estimates system states, such as voltage magnitude and angle, is equipped with bad data detection schemes to filter out bad data. Hence, for a successful attack, it has to bypass the bad data detector to remain undetected. In order for the bad data detector to not eliminate the false data, the state estimation residue resulting from falsified data should be smaller than the state estimation threshold. In addition to the attack on PMUs, it is also assumed that the attackers send fabricated relay signals to the control center, to conduct a coordinated

attack. This is due to the fact that in an actual line fault event, the relay will send a trip signal to the control center. In order to mimic the real fault, attackers have to compromise the relay to send fabricated line tripping signals to the control center. With this coordinated attack strategy, attackers can prompt the operator to take wrong actions which jeopardize system reliability. Since it is difficult to compromise all PMUs, it is assumed that the attacker can only gain access to a limited number of PMUs.

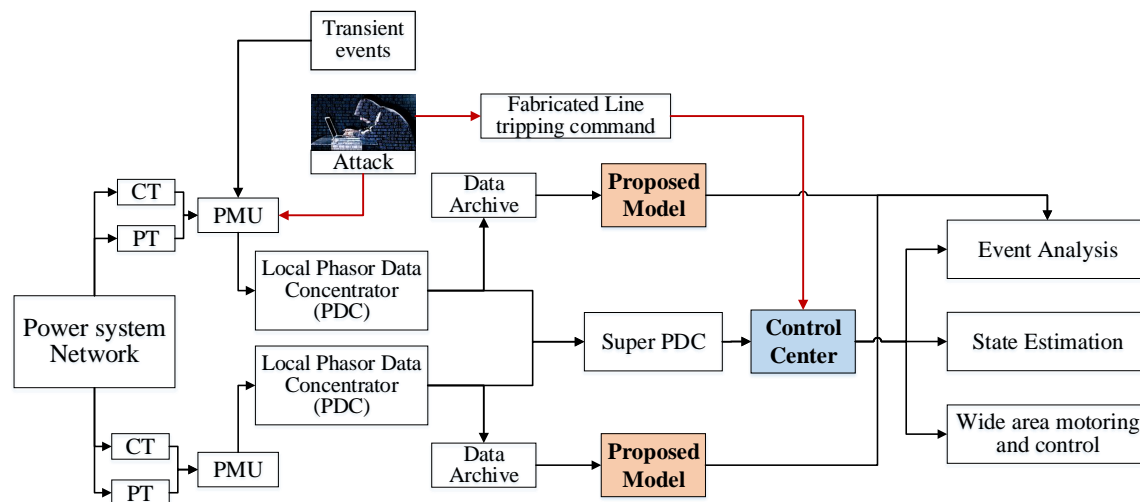


Figure 2.5: Architecture of the data-driven model in power systems

2.5.2 False Data Creation

Three data spoofing strategies, namely data drop attack, playback attack, and time attack are used to simulate false data attacks. Data spoofing is an attack strategy where the spoofed data becomes similar to a normal data, which makes detecting cyber-attacks a challenging task [30]. Data drop attacks directly alter data values to falsify data. During a playback attack strategy, attackers can use a historical PMU data to replay a transient event [22]. In a time attack strategy, the post-event PMU data is re-sampled at a different sampling rate [31]. Once PMU communication network is compromised by the attackers, these three data spoofing methods can be used to alter time series data sampled by PMUs. In this chapter, it is assumed that the data drop attack randomly alters the magnitude of the

data from 10% to 40%, while the playback attack mimics a fault transient to replace the original data to misinform the operators. In a time attack, it is assumed that attackers alter the sampling frequency of the fault transients. The re-sample rate of the time attack ranges from a factor of 1.5 to 3, slower than the actual sampling rate of PMUs. Fig. 2.6 illustrates an example of the three types of data spoofing strategies where seven bus voltages and 28 current magnitudes are plotted. The length of the voltage and current waveforms are 1.5s, i.e. 90 data points to match PMU sampling rate of 60 frames per second. It can be seen that several voltage and current measurements are directly altered in a data drop attack, while in a playback attack, a fake transient event signal is injected to replace real PMU data. In a time attack, the post-event voltage and current measurements appear stretched because the sampling frequency is changed by the attacker.

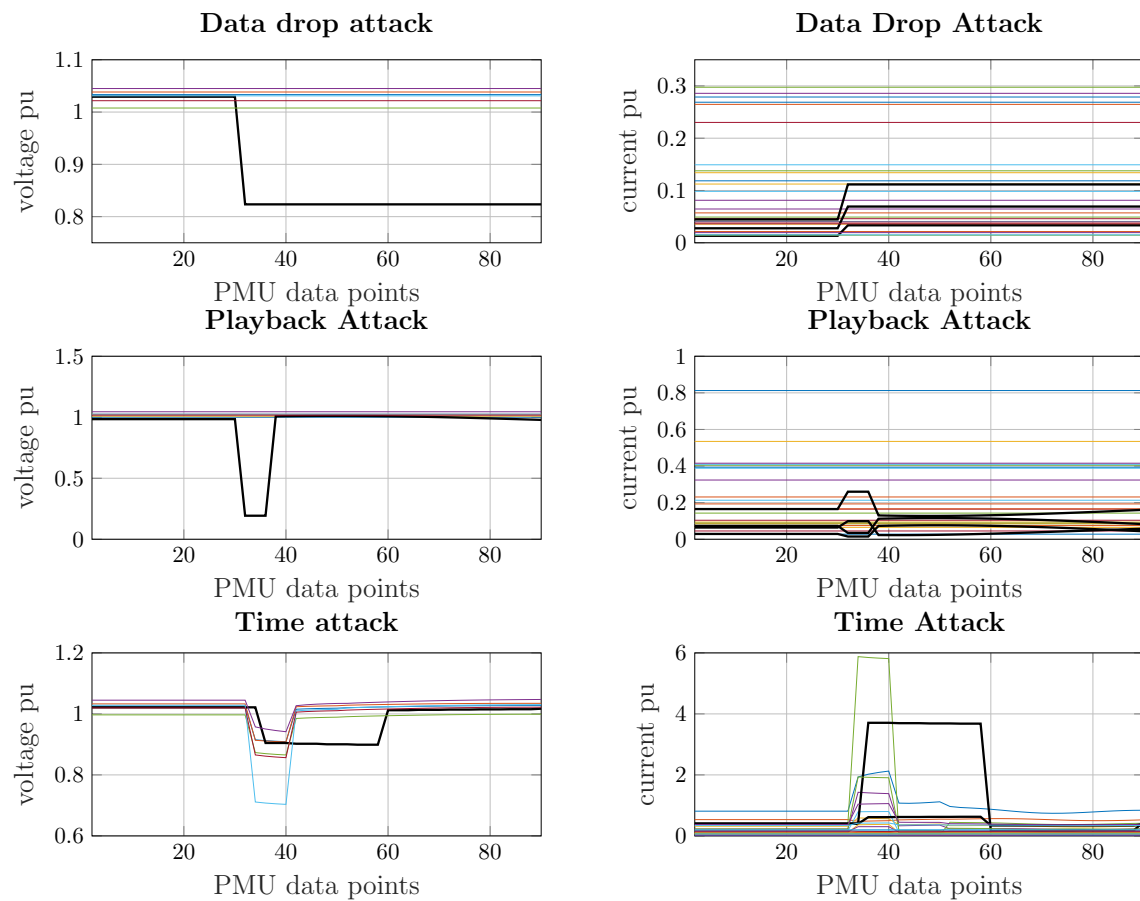


Figure 2.6: Voltage and current waveforms for three types of data spoofing attacks

2.6 Simulation Results

2.6.1 PMU placement

Although the number of PMU installations in power systems has increased in recent years, existing power systems do not possess PMUs in every single location. Therefore, to study a practical power system, it is assumed that the studied power system is observed by both PMU and SCADA measurements and no measurement redundancies exist. The placement of PMUs in IEEE 30-bus and IEEE 118-bus systems are given in Table 4.1 following the placement algorithm introduced in [38] that guarantees power systems are fully observable with the available measurements.

Table 2.4: PMU placement of IEEE 30-bus system and IEEE 118-bus system

Test System	PMU placement
IEEE 30-bus system	2, 4, 6, 10, 12, 15, 27
IEEE 118-bus system	5, 12, 15, 17, 32, 37, 49, 56, 59, 67, 69, 70, 71, 77, 80, 85, 92, 96, 100, 105

As the size of IEEE 30-bus system is small, it is assumed that only one PDC is used to aggregate seven PMU voltage and current data. In the IEEE 118-bus system, the power grid is divided into four zones that are given in Fig. 2.7, where each zone has one PDC that gathers 26 PMU time series data from each local area.

2.6.2 Event Data Generation

When the continuously recorded PMU data, i.e., voltage, frequency or rate of change of frequency, exceeds a pre-defined disturbance detection threshold, the disturbance data will be archived in a separate location in PDC for post-event analysis [8]. North American

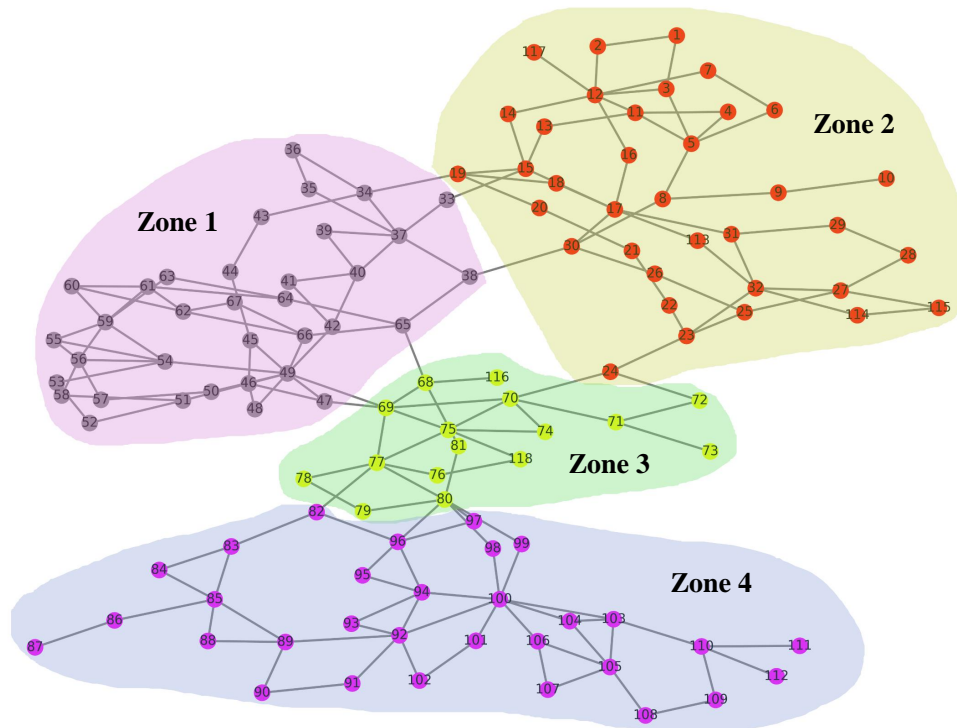


Figure 2.7: Four zones for IEEE 118-bus system

Electric Reliability Corporation (NERC) has defined the threshold for detecting transient events and has set a requirement of the archived disturbance data to have at least 30 cycles including at least 2 cycles before any event [39]. Therefore, the archived PMU voltage and current disturbance data for the duration of 1.5 s (0.5 s of pre-event and 1 s of post-event data) are used to classify transient events and false data. To evaluate the efficiency of the PMU-based data-driven approach, fake transients due to false data attacks and five types of real transient events, namely, faults, generation change, line tripping, load change, and shunt switching are investigated under four different scenarios:

1. System with peak loading condition;
2. System with light loading condition;
3. System with light loading condition and one generator out of service; and
4. System with light loading condition and one line outage.

False data is generated based on the aforementioned data spoofing strategies. Three-phase balanced fault, single line to ground fault (on phase A) and double line to ground fault (on phase A and B) are simulated on each transmission line, at a random selected distance to one side of the line. All three types of faults are cleared after 5 cycles to simulate a real fault clearing mechanism. It is to be noted that this study uses phase A voltage and current data to evaluate the proposed approach. If a single line to ground fault or double line to ground fault occurs on other phases, that is, phase B and phase C, the developed approach that utilizes voltage and current data from the other phases can still correctly classify the event. This is due to the fact that faults that occur at different phases exhibit similar oscillation patterns in voltage and current data. A random generation and load change are simulated between 10% to 40% of their nominal values. Line tripping is simulated on each transmission line to mimic a scheduled maintenance. Shunt switching is also simulated to mimic a switched shunt control. All transient events are simulated in TSAT, a power system transient stability simulation software [40].

The number of each simulated power system event is given in Table 2.5. An example of voltage and current waveforms in the vicinity of the five types of real transient events are given in Fig. 2.8. The length of each waveform is 1.5 s, i.e. 90 points. It is worth noting that, for each event, the shape of the voltage and current waveforms at different locations resemble each other while the variations are different. For each event, all PMU bus voltage and branch current data are used to construct the M_{BOP} voltage and current matrices, and are further processed and converted to weighted matrices using the TF-DF algorithm.

Table 2.5: Number of studied events

Event Type	Number of Events	
	IEEE 30-bus system	IEEE 118-bus system
Fault	964	1480
Generation change	161	360
Line tripling	163	680
Load change	504	728
Shunt switching	120	224
False data	600	1200
Total	2512	4672

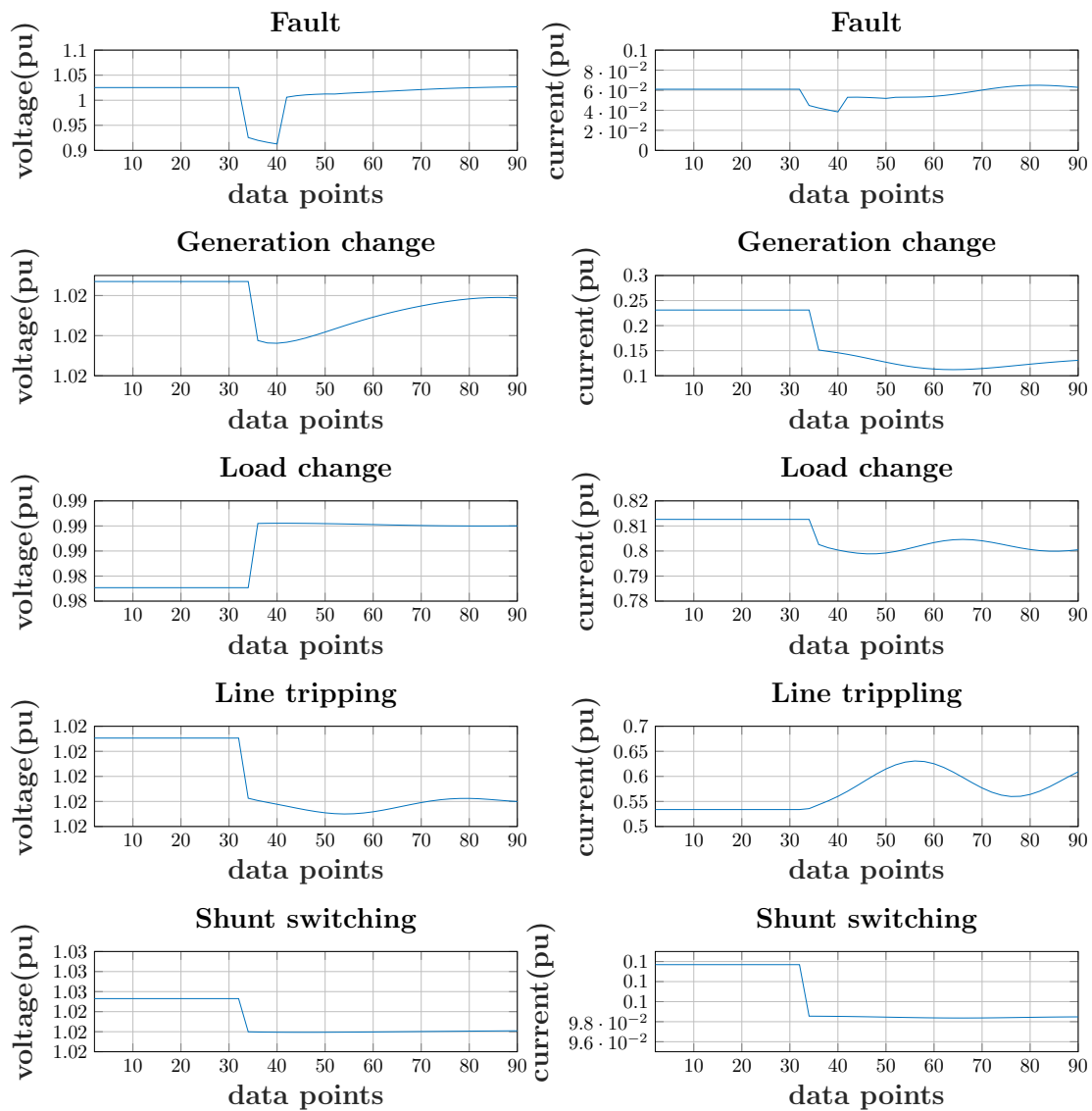


Figure 2.8: Voltage and current waveforms in the vicinity of various event types

2.6.3 Model evaluation

After extracting features from the labeled multivariate PMU time series data, SVM and Ensemble classifiers using Bagged Trees are used to recognize the aforementioned six types of events. For the IEEE 30-bus and IEEE 118-bus systems, the size of the multivariate time series data for each local zone is 35 (28 current magnitudes and 7 voltage magnitudes), and 26 (21 current magnitudes and 5 voltage magnitudes), respectively. Both voltage and current data are used to classify faults, generation change, load change, line tripping, shunt switching, and false data attacks. Since a local PDC gathers measurements from a limited number of PMUs in a zone, the proposed approach can be utilized to classify events in a substation level and is independent of the size of a power network. Therefore, the proposed approach can be readily applied to larger systems.

The impact of SAX parameters on the classification accuracy is investigated by varying the size of the alphabet α , word γ , and the length of the sliding window ω . The classification accuracy, which distinguishes fault, generation loss, line tripping, load change, shunt switching, and fake transients caused by false data attacks, is evaluated after feeding the feature vectors into the aforementioned classifiers. Classification accuracies for IEEE 30-bus system under different combinations of SAX parameters are given in Fig. 2.9. The cross-validation number that is used to split data into training and validation sets, is selected as 10, which is common among data mining techniques [41]. The classification results presented in Fig. 2.9 show that the developed PMU-based data-driven approach performs well when classifying the studied six types of events. The proposed method achieves 99.7% accuracy on this 1.5s data when the alphabet size is 3, size of the word is 4, and the length of the sliding window is 60.

The classification confusion matrix given in Fig. 2.10 shows that all generation change, line tripping, and shunt switching are correctly classified. One percent of misclassification between false data attacks and faults is due to the similarity in voltage and current streams

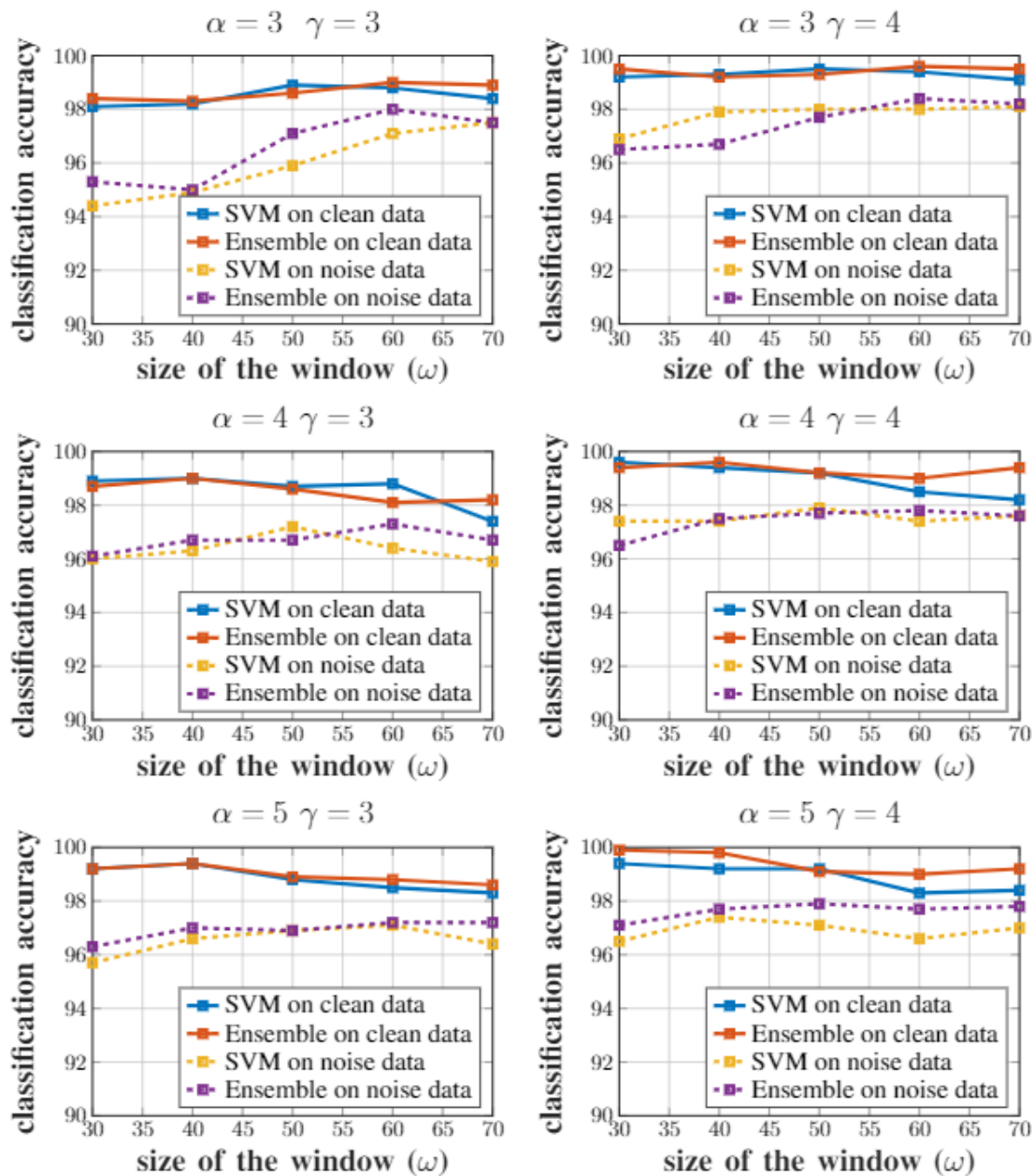


Figure 2.9: Average classification accuracy (in percent) of six events for different SAX parameters for IEEE 30-bus system

between faults and time attacks, when the re-sampling rate of time attack is close to real sampling rate of PMUs. One percent of load change events are misclassified as generation changes, due to similarity between transient signals for large load fluctuations and generation changes.

False Data	99%	<1%				
Fault	<1%	99%				
Generation Change			100%	<1%		
Load Change				>99%		
Line Tripping					100%	
Shunt Switching						100%
	False Data	Fault	Generation Change	Load Change	Line Tripping	Shunt Switching
Positive Predictive Value	>99%	>99%	100%	>99%	100%	100%

Figure 2.10: Confusion matrix for the developed classification method for the IEEE 30-bus system (accuracy = 99.7%, $\alpha = 3$, $\gamma = 4$, $\omega = 60$)

2.7 Discussion

2.7.1 Impact of Model Parameters

Further analysis of the impact of SAX parameters, i.e. α , γ and ω , on the classification performance in Fig. 2.9 shows that increasing α would slightly increase the classification accuracy. For instance, given $\gamma = 3$, $\omega = 30$ and the SVM classifier, increasing α from three to five could improve the classification accuracy from 98% to 99.2% for clean data and 94.2% to 96% for noisy data, respectively. It is also seen that larger γ leads to a higher classification accuracy. Given the same ω and α , increasing γ from three to four can enhance the classification accuracy by 2%. Increasing the length of the sliding window, ω , can improve the classification accuracy on noisy PMU data. For example, given $\alpha = 3$,

$\gamma = 3$ and using the SVM classifier, the classification accuracy changes from 94.2% to 97.5% for noisy data when changing ω from 30 to 70. However, for clean data, increasing ω could slightly decrease the performance of the developed data-driven model, especially when ω is larger than 40 and α is larger than three. It can be seen in Fig. 2.9 that the classification results drop from 99.3% to 98.1% when ω changes from 40 to 70 and $\alpha = 5$, $\gamma = 3$ and the classifier is SVM.

The classification results in Fig. 2.9 show that increasing α and γ leads to an improvement in accuracy. However, there is a trade-off between the size of the feature vector and the size of α and γ . Large α and γ will generate a large feature vector and thus lead to a longer computation time for classification. This is not desired for dynamic visualization applications, where the time it takes to classify events should be minimized. Comparing the accuracy and the size of the feature vector, a combination where the $\alpha = 3$, $\gamma = 4$, and $\omega = 60$ is chosen for classifying events. With this choice, the classification accuracy reaches 99.7% for clean data. It is observed that Bagged Trees Ensemble classifier achieves a higher classification accuracy than SVM by comparing the classification accuracies for both clean and noisy data in Fig. 2.9. For the IEEE 118-bus system, with Bagged Trees Ensemble classifier and the suggested SAX parameters, the classification accuracy reaches 97.2% and 96.1% for clean and noisy data, respectively. The confusion matrix for the IEEE 118-bus system for clean data is given in Fig. 2.11. It can be observed that nearly 99% of false data, 97% of faults, 98% of generation change, and 98% of load change are correctly classified. The misclassification between line tripping and shunt switching can be explained by the fact that these two events both belong to network operations [40].

False Data	99%	1%				
Fault	1%	97%	1%		1%	
Generation Change		<1%	98%	1%		
Load Change			<1%	98%		
Line Tripping		1%		1%	93%	4%
Shunt Switching		<1%			6%	96%
	False Data	Fault	Generation Change	Load Change	Line Tripping	Shunt Switching
Positive Predictive Value	99%	97%	98%	98%	93%	96%

Figure 2.11: Confusion matrix for the developed classification method for the IEEE 118-bus system (accuracy = 97.2%, $\alpha = 3$, $\gamma = 4$, $\omega = 60$)

2.7.2 Impact of the Coverage and Placement of PMUs

To evaluate the effectiveness of the developed methodology under different number of PMUs, sensitivity studies, varying the number of PMUs as well as their location, are also performed. The results of this study on the IEEE 30-bus and IEEE 118-bus systems are presented in Table 2.6. These results show that the developed data-driven approach achieves a high classification accuracy with different coverage rates of PMUs, although the classification accuracy is slightly decreased with fewer number of PMUs. It is also found that different placements of PMUs do not have a significant impact on the classification accuracy of the developed approach.

Table 2.6: Classification accuracy under different coverage and different placement of PMUs ($\alpha = 3, \gamma = 4, \omega = 60$)

System	Different coverage of PMUs		Different placement of PMUs	
	Strategy	Classification accuracy (%)	Strategy	Classification accuracy (%)
IEEE 30-bus system	5 PMUs	99	5 PMUs	99
	6 PMUs	99.3	5 PMUs	98.2
	7 PMUs	99.7	5 PMUs	98.4
IEEE 118-bus system	12 PMUs	95.4	12 PMUs	95.4
	16 PMUs	95.8	12 PMUs	95.1
	20 PMUs	97.2	12 PMUs	95

2.7.3 Robustness to Noisy Measurements

The amount of noise present in the PMU field data decreases the event classification accuracy. Thereby, it is important to either eliminate the noise or guarantee robustness of classification methods to noise. As indicated in [8], it is hard to filter out noise due to similarity between noise and power system fluctuations such as load change. This emphasizes the necessity to investigate robustness of the proposed data-driven approach against noise. Xie et al. in [42] propose adding a noise with around 92 dB signal to noise ratio (SNR) to the simulated PMU data so that the combined data resembles a field PMU data. As a result, a white Gaussian noise with 90 dB SNR is injected to all studied event data, for both test systems. The proposed method achieves a 98.2% and 96.1% classification accuracy with the previously suggested SAX parameters for IEEE 30-bus and IEEE 118-bus systems, respectively. Intuitively, the mean value computed from each sub-sequence reduces the sensitivity to white Gaussian noise in the SAX method.

2.7.4 Performance with Untrained False data

Since the work presented in this chapter is based on a supervised machine learning algorithm, for the unlabeled attack scenario, the classification accuracy might be decreased. In order to demonstrate the performance of the developed data-driven model when the adversary uses an attack not considered during the training, a different playback attack strategy from that used in the offline training phase is also considered. This new playback attack strategy mimics the transients that occur after a load change in a power system, as illustrated in Fig. 2.12. With the offline-trained data-driven model, a total number of 500 attack events obtained from this new attack strategy is utilized to evaluate the developed model. The data-driven model successfully identifies all 500 fake events for both the IEEE 30-bus and IEEE 118-bus systems when the data is clean. For the noisy PMU data, the developed model successfully classifies 488 attack events as fake events for IEEE 30-bus system, and all 500 fake events for the IEEE 118-bus system. Comparing the correlation between the multivariate PMU data of the attack events with that of the real events, helps explain the resulting high classification accuracy for unlabeled scenarios. This comparison shows a low correlation between the multivariate PMU data during attacks, as compared to a real event. This low correlation is due to the fact that the attackers can only alter a limited number of PMUs. These results corroborate the fact that considering the correlations among PMU data is the key to more accurate identification of system events.

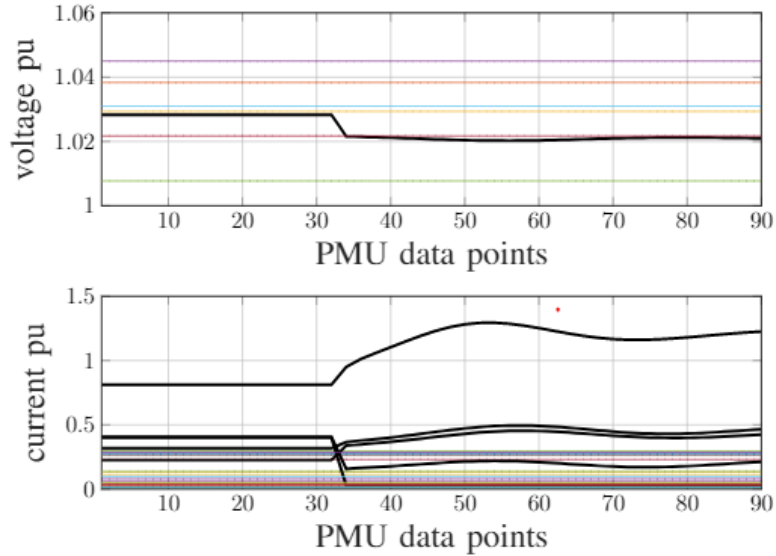


Figure 2.12: An attack strategy not considered during the training

2.7.5 Comparison with Conventional Event Classification Methods

Classification accuracies of the traditional pattern recognition methods such as Discrete Wavelet Transform (DWT), Principal Component Analysis (PCA), and more recent methods such as Domain Specific Shapelet (Dshapelet) and Slope of Dshapelet (S^3), are compared with the developed multivariate BOP method. DWT uses the mother wavelet, which is a set of basic functions decomposing the data into multiple components, to transform the time series data into several resolution levels [43]. The coefficients, which represent the detailed and approximate information of the data, can be used as features for pattern recognition [43]. By comparing the three widely used wavelets, i.e. Daubechies wavelet, Least Asymmetric wavelet and Coiflets wavelet [43] for classifying transient events, the mother wavelet Daubechies is selected for DWT method, as this wavelet reaches better classification accuracy among all. All approximate and detailed coefficients from DWT are selected to differentiate each event. PCA projects the data onto the principal subspace, which is a lower dimensional linear space, such that the variance of the data is maximized [44]. The dynamics of the data can be analyzed by transferring the data into

a combination of Principal Components (PCs) [16]. Time series can be expressed to a set of PCs where the first PC represents the largest variance of the data. In this study, by comparing the classification accuracy after using different number of PCs, the 15 top largest PCs are selected as features to distinguish each event. Shapelet is the subsequence of a time series that can maximally represent a time series [17]. Authors in [8] and [16] utilize Dshapelet and S^3 to extract the shapelet of the time series data and the one-step slope sequence of the shapelet as the feature vector. Dshapelet identifies a unique shapelet that can represent the most dramatic change while S^3 captures the trend of the most dramatic change [8, 16]. Ensemble classifier based on Bagged Trees is used to compare these traditional methods using clean and simulated field data. Classification accuracies given in Fig. 2.13 indicate that the developed data-driven approach leads to a better accuracy when classifying the aforementioned six types of events, and is more robust against noise. Feature extraction time for PCA, DWT, Dshapelet, S^3 and multivariate BOP methods are 0.009s, 0.005s, 0.0015s, 0.002s and 0.026s respectively, on a computer with i7-7700 CPU, 4.2GHz core and 32GB RAM. In sum, the run-time for the second phase of the event classification in Fig. 2.1, which occurs online, is approximately 1.654s. This includes 1.5s for event data collection, 0.026s for feature extraction, 0.028s for classification, and 0.1s for communication delays [8], which is a reasonable time for near real-time applications.

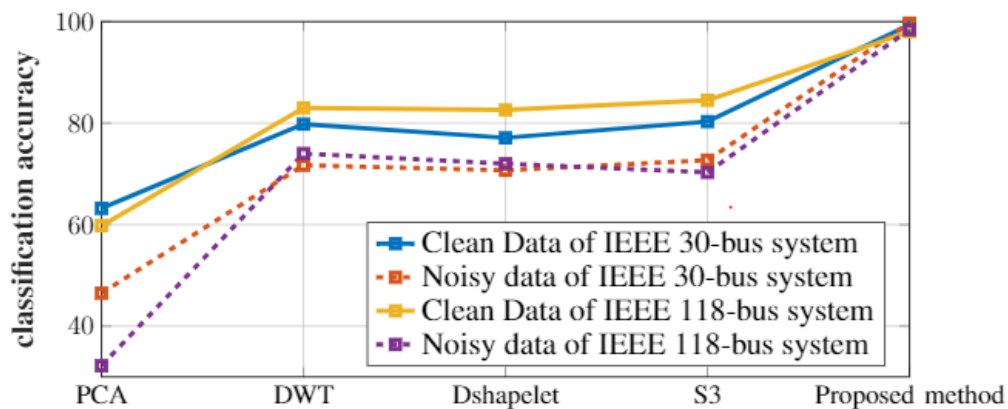


Figure 2.13: Classification accuracy for both clean and noisy data

2.8 Conclusions

False data attacks on PMUs pose major threats to the PMU-based dynamic visualization tools, as *counterfeit* events might lead to wrong identifications of system events, which in turn prompt ineffective actions that may compromise system reliability. It is thus crucial to guarantee the credibility of the dynamic information obtained from the visualization tools. This chapter introduced a new data-driven approach for classification of transient events in power systems, that is based on data received from PMUs. The key distinction between this method and the existing state-of-the-art is the ability to distinguish false-data attacks from the other transient events. Three different types of false data attacks, namely, data drop attack, playback attack, and time attack are considered to evaluate the effectiveness of the developed approach. Power system event oscillation patterns from a large volume of data are extracted using an efficient SAX-based methodology, while event classification is carried out using an effective machine learning classifier, i.e. Bagged Trees Ensemble classifier. Case studies show that the data-driven approach is superior to the other classical methods in classifying fake transients caused by false data attacks and real transient events, in terms of classification accuracy and robustness against noise. The promising performance of the developed method on this 1.5s time series data in classifying six different types of power system events makes it possible to provide fast, accurate and reliable dynamic information to system operators to facilitate post-event decision-making process and prevent potential cascading failures. Additionally, as the developed data-driven approach is applied at regional PDCs installed at each substation, the developed event classification approach can be utilized in large power systems where thousands of PMUs are deployed.

Chapter 3

Impact Prediction of Power System

Events in Steady State

3.1 Introduction

When a system event, such as bus and line faults, occurs, some system components may be failed, that in turn will cause further system failures. This phenomenon is the cascading failure, where the failure of one or more components leads to the failure of others and so on. The objective of this chapter is to predict the possible propagation of cascades, and inform system operators of the critical system components. Furthermore, as the unprecedented increase in power system uncertainties, due to factors such as increased penetration of renewables and load uncertainties, makes real-time power system conditions less predictable, a dynamic cascading failure prediction model, that is adaptive to different operating conditions, is developed to predict the most probable cascade sequence.

The developed model enables fast estimation of the interactions between system components given present power flows. Concretely, the model first extracts failure propagation patterns from a large-scale simulated cascade data under different grid operating conditions. Next, specific interactions that correspond to the present transmission line power flows can be generated that in turn can be used to forecast potential subsequent failures in cascades under up-to-date conditions. Having the knowledge of the potential failures, operators could block the corresponding relays [45], i.e., stop tripping of the components and still keep these components in service in power systems, to buy time to identify appropriate

mitigative actions and prevent or reduce the impact of cascades.

3.2 Background

The concept of developing a cascading failure model has been given significant attention in the power engineering research, and a variety of cascading failure models have been developed. Examples include the ORNL-PSerc-Alaska (OPA) model [46–48], Manchester model [49], Hidden failure model [50], topology-based model [51–55], CASCADE model [56] and the branching process model [57] to simulate and analyze the impact of cascading failures on power systems. The developed failure models can be grouped into three different categories: (1) *topology-based* [51–55], (2) *DC/AC-based* [46–50, 58–60], and (3) *statistical* models [56, 57]. Topology-based models, motivated by the complex network methods [61–63], represent power systems as a large-scale network in which failures spread from a node to its neighbors. With a proper network representation, various methods such as centrality measures (e.g., betweenness) can be used to assess a system’s vulnerability to various failures. What limits the application of topology-based methods is the fact that a cascading failure is only treated as a local phenomenon, i.e., loss of a node only affects the neighboring nodes. However, in a real power system, cascading outages propagate globally [64]: a failure of one component can cause another failure in a distant location. Additionally, it is demonstrated in [65] that these topological cascading failure models could lead to erroneous conclusions regarding vulnerable system components, which may result in wrong mitigation strategies. DC/AC-based models such as the OPA model and the Manchester model address this issue by modeling failures with a power law distribution. Based on the DC/AC power flow, the propagation of cascades can be simulated to gain insights into the physics of cascade propagation in the power grids [66]. Hence, the information of cascading outages at each point in time (such as power flows and load losses) can be used to understand the propagation and impacts of cascades [60].

These models, however, are computationally expensive and generate large-scale data [66]. Statistical models, such as the CASCADE model and Branching process model, enable fast generation of cascade data, but neglect power system information such as topology, power flows and power injections [56,57]. While these models reveal the cascade size, the contribution of each component to a cascading failure, a necessary information for online operations, e.g., generation ramping, cannot be derived. To address the drawbacks of the statistical models, data-driven statistical models such as the influence graph model [66,67] and the interaction model [68–70] are developed to simulate cascading failures using the *interactions* among system components. An interaction is defined as the probability of one component’s failure, given the failure of other components. This failure probability can be estimated from historical cascading failure data or data generated from the DC/AC-based models. These interaction and influence graph method, however, do not predict potential failures in near real-time. Recent studies have used learning-based method to efficiently analyze cascading failures. Authors in [71] propose to apply reinforcement learning to efficiently identify critical fault chains. In another study [58], Artificial neural network (ANN) is used to promptly estimate Energy-Not-Supplied in real time. However, the failure probability of each system component, an important information for system protection, cannot be derived.

3.3 AC-based Cascading Failure Model

The IEEE Power and Energy Society Cascading Failure Working Group (CFWG) has defined a cascading failure as a sequence of dependent component failures in which the failure of one or more component leads to the failure of others, continuing with further subsequent failures [72]. An example of the cascade data is shown in Fig. 3.1, where the initial event in generation 1 produces two subsequent failures (i.e., two *children*) in generation 2. The failure of these two components (lines 4 and 5) causes line 2 to fail in

generation 3 due to overloads. As no more components are outaged after the outage of line 2, this cascade stops after three generations.

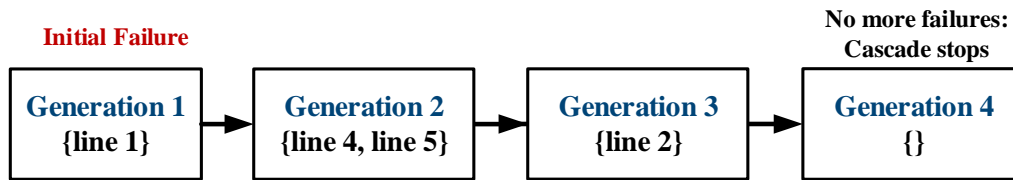


Figure 3.1: Example of a cascade propagation

As the cascading failure events in practical is rare and confidential, it is hard to have sufficient cascade data for analysis. Hence, to generate cascade data, a simulation based model, i.e., AC OPA model is used. This model has been previously validated in [48, 73]. AC OPA model uses AC optimal power flow (AC OPF) to solve the power flow and determine the operator actions such as load shedding. The detailed steps followed for the AC OPA model used in this chapter are as follows:

- Step 1 Each load is initialized by multiplying a random number that is uniformly distributed in $[2 - \gamma, \gamma]$ to the nominal value of the load, where γ is the load variability and is set to 1.67. Notice that all loads vary asynchronously in this study. The choice for the load variability γ is inspired by [68, 69] to represent load variations throughout a year. The load and generation profiles of the two test systems can be obtained from [74] and [75];
- Step 2 Line power flows are initialized using AC OPF. If AC OPF diverges, loads are shed until a solution is reached. If no solutions are obtained, back to step 1;
- Step 3 An initial event is set. It is assumed that each system component fails independently with a failure probability of 0.001, as the initial events are rare in practice. Notice that the initial failure probability can be further improved by analyzing the historical failure data;

Step 4 All the islands within the system are identified. For each island, the supply and demand are balanced by ramping up/down generation or shedding load. After the generation and load in all islands are re-balanced, the AC OPF is calculated. If AC OPF diverges, loads are shed until a solution is reached;

Step 5 If any line flows violate their limits, the overloaded lines are tripped and Step 4 is repeated; otherwise, the simulation will stop.

3.4 Dynamic Failure Model

3.4.1 Overview of the Dynamic Failure Model

Here we first give an overview of the developed dynamic failure model. The dynamic failure model is developed to enable (1) fast estimation of failure patterns under online operating conditions, and (2) to predict the propagation of cascades, as illustrated in Fig. 3.2. Offline analysis uses historical or simulated cascade data under different operating conditions to learn all the failure interactions among components under different system states. The real-time analysis, using PMU measurements, identifies only those interactions (among all) that apply to real-time conditions, hence referred to as *dynamic interactions*. Once the updated interaction model is generated, a methodology developed in this work, referred to as the *propagation tree method*, is used to predict potential propagation of cascades. Eventually, this model helps power system operators better understand the outcomes of the potential failures under the present system operating condition.

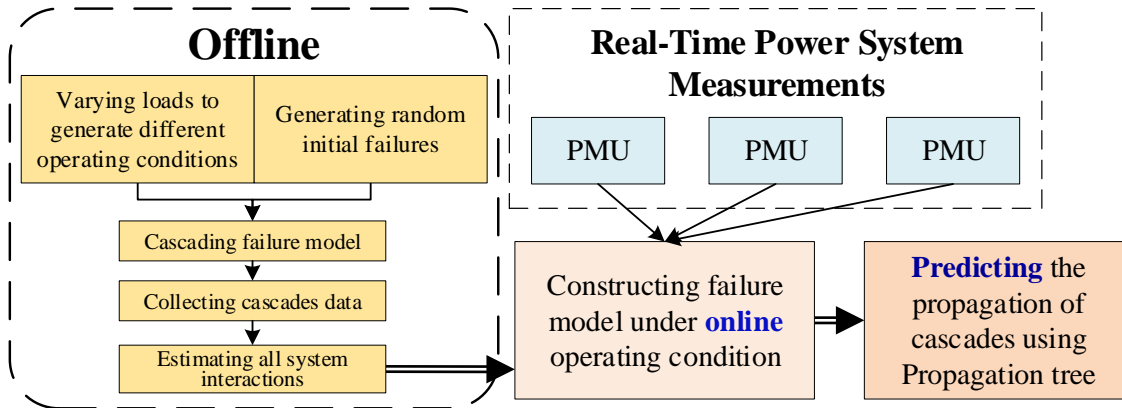


Figure 3.2: Development of a dynamic failure model for predicting cascades

3.4.2 Estimating the Dynamic Interactions

An interaction is defined as the probability of one component's failure, given the failure of other components. Conventional interaction estimation model is firstly introduced in [68] to identify critical system components that have large contribution to cascading failures. The same authors have further improved the conventional interaction model in [69] to use an efficient estimation technique, i.e., Expectation Maximum (EM) algorithm, to estimate component failure interactions with less cascade data. While the main idea of calculating failure interactions is the same, the difference is in the estimation algorithm. Although conventional interaction models can capture the propagation patterns from cascade data, they neglect how component failure interactions vary with different loading levels of system components. Intuitively, a transmission line with a higher loading ratio of $\frac{\text{Power Flow}}{\text{Thermal Capacity}}$ is more likely to fail compared to a line with a lower loading ratio. To capture the effect of line loading on the propagation of cascades, we develop a dynamic interaction model that incorporates the system loading conditions into the failure interactions. Hence, the model enables updating the interactions based on the up-to-date power flows. A dynamic interaction between failures of two components can be thought of as a

conditional failure probability $P(j | i, r_j)$:

$$P(j | i, r_j) = P(j \text{ fails in generation } k + 1 | i \text{ fails in} \\ \text{generation } k, \text{ loading ratio of line } j), \quad (3.1)$$

where $P(j | i, r_j)$ is the probability of line j failing, given that line i has failed and loading ratio for line j is r_j . The loading ratio (or state) r_j for line j is defined as,

$$r_j = \frac{pf_j}{pf_j^{\max}} \quad (3.2)$$

where pf_j and pf_j^{\max} are the apparent power flow and maximum apparent power flow capacity of line j . With the obtained cascade data under different operating conditions, Bayes' theorem is used to estimate the conditional probability $P(j | i, r_j)$ [76],

$$P(j | i, r_j) = \frac{P(j, r_j | i)}{P(r_j | i)} \quad (3.3)$$

where $P(j, r_j | i)$ is the joint probability of the failure of line j and the state of line j , given line i has failed in the previous generation. $P(r_j | i)$ is the conditional probability of the state of line j given the failure of line i . In this study, the loading conditions of system components in a cascade are the states before the occurrence of the initial failures, as in a statistical model it is infeasible to predict the changes in the loading conditions during the propagation of cascades.

To estimate the conditional failure probability $P(j, r_j | i)$ in (3.3), the methodology introduced by Qi et al. [68] is adopted. Under a specific loading ratio r , an *interaction matrix* $A \in \mathbb{R}^{n \times n}$ can be obtained, where n is the number of system components. The interaction matrix summarizes the interactions between system components using the stages of cascade data. The elements of A , A_{ij} , represent the number of times, among all generations

of all cascades, component j fails subsequent to the failure of component i in the previous generation, when the loading ratio of component j is r_j .

The conventional interaction model in [68] assumes that the failure of component j in a generation is only subsequent to one component's outage in the previous generation; and that component is the one that has led component j to fail the most among all the cascades. This is due to the difficulty in identifying the *complete* cause of subsequent outages in cascade data. After obtaining A , the empirical failure interaction probability between two failed components can be computed and recorded in an *interaction probability matrix* B , with its elements $B_{ij} = \frac{A_{ij}}{n_i}$. B_{ij} is the interaction probability between lines i and j , i.e. the empirical joint failure probability of the state of line j and the failure of line j caused by line i , and n_i is the number of times component i fails among all cascades. The probability matrix B quantifies the interaction between any two lines during the propagation of cascades under a specific line loading ratio. B_{ij} can be thought of as the conditional failure probability $P(j, r_j | i)$.

The initial failures in the first generation are often caused by exogenous events, e.g., a tree falling on a line, while failures in the subsequent generations are caused by the outages of other system components [77]. Hence, the dynamic interaction between two components is separated into two different interactions: (1) initial interaction $P_0(j | i, r_i, r_j)$ and (2) subsequent interaction $P_{1+}(j | i, r_j)$, defined as

$$P_0(j | i, r_i, r_j) = \frac{P_0(j, r_j, r_i | i)}{P_0(r_j, r_i | i)}, \quad (3.4)$$

$$P_{1+}(j | i, r_j) = \frac{P_{1+}(j, r_j | i)}{P_{1+}(r_j | i)}, \quad (3.5)$$

where 0 and 1+ denote the initial generation and subsequent generations of cascades, respectively. The initial interaction is extracted from the outaged components between the first and second generations, while the subsequent interactions are obtained from all

cascade generations except the first generation. The benefits achieved by dividing the initial and subsequent interactions separately will be discussed further in Section III-C. For simplicity, we use D to denote *dynamic* interaction matrices, and D_0 and D_{1+} to denote the initial and subsequent interaction matrices for the rest of the chapter.

It can be observed from (3.2) that the loading ratio r_j is in range $[0, 1]$. It is impractical to estimate the failure probability of line j for any r_j , as r_j is a continuous value. Therefore, instead of estimating failure probability $P_0(j | i, r_i, r_j)$ and $P_{1+}(j | i, r_j)$ for any loading ratio r , based on Bayes' theorem, we discretize the two interaction probabilities as [76],

$$P_0(j | i, r_i \in [\frac{n}{b}, \frac{n+1}{b}), r_j \in [\frac{m}{b}, \frac{m+1}{b})), \quad (3.6)$$

$$P_{1+}(j | i, r_j \in [\frac{m}{b}, \frac{m+1}{b})), \quad (3.7)$$

where b is the number of bins used to discretize continuous variable r and $m, n \in \{0, 1, \dots, b-1\}$. For instance, if we use two bins, i.e., $b = 2$, there are four initial interactions: $P_0(j | i, r_i \in [0, 0.5), r_j \in [0, 0.5))$, $P_0(j | i, r_i \in [0, 0.5), r_j \in [0.5, 1.0))$, $P_0(j | i, r_i \in [0.5, 1.0), r_j \in [0, 0.5))$ and $P_0(j | i, r_i \in [0.5, 1.0), r_j \in [0.5, 1.0))$. Also, there are two subsequent interactions: $P_{1+}(j | i, r_j \in [0, 0.5))$ and $P_{1+}(j | i, r_j \in [0.5, 1.0))$.

3.4.3 Predicting the Cascade Propagation

In the previous section, the dynamic interactions between system components were estimated and a methodology to incorporate online measurements into these interactions was introduced. To predict potential cascades online, we develop a methodology to calculate the failure probability of each component in each generation of a cascade. In this study, as the causes of the initial events are exogenous, we only focus on analyzing the propagation of cascades. Calculation of failure probabilities is conducted by constructing a *Propagation Tree*, an example of which is shown in Fig. 3.3. Each layer of the tree corresponds to

a generation of a cascade, and each node in a layer represents a component that might fail in the corresponding generation. An edge (i.e., branch) in a propagation tree denotes the interaction between two nodes in two consecutive layers. When the failure probability of a component in a generation is lower than a threshold ε , it is assumed that this component would not generate subsequent child failures. Thus, the corresponding node in the propagation tree becomes a *leaf node*, such as node 3 in generations 3 and 5 in Fig. 3.3. The steps to construct a propagation tree are:

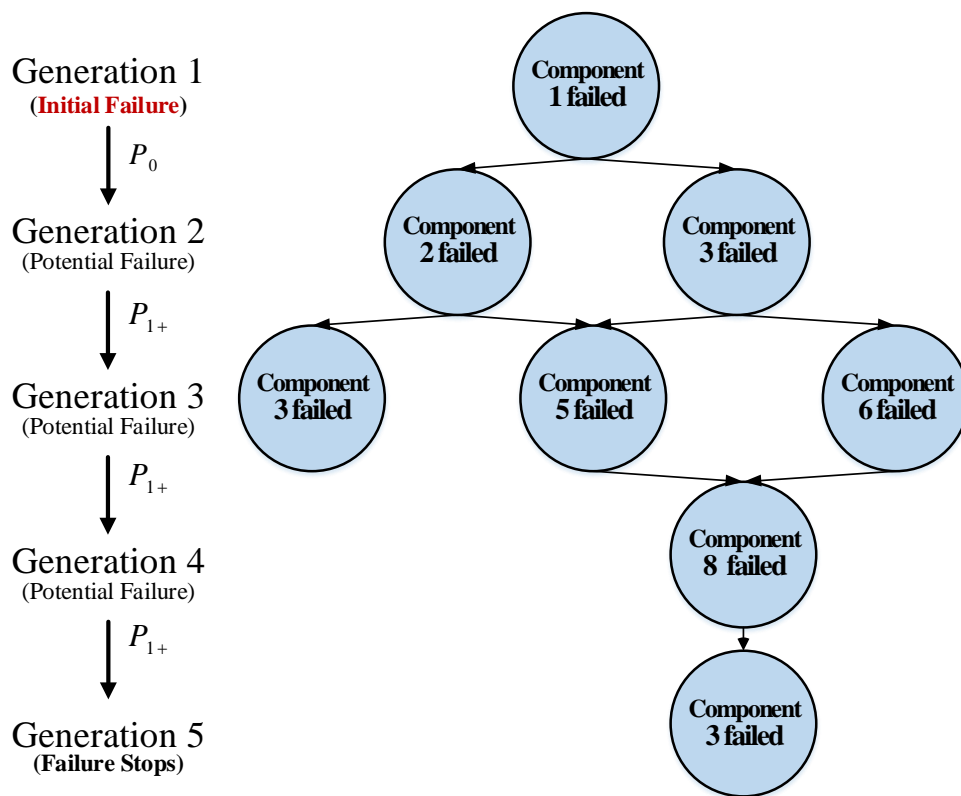


Figure 3.3: An example of a propagation tree

Step 1 Given the online operating conditions of each system component, $r_q, q \in \{1, 2, \dots, n\}$, where n is the total number of system components, the corresponding initial P_0^{current} and subsequent P_{1+}^{current} interactions in (3.6) and (3.7) are found among the interactions studied offline;

Step 2 Given the initial failed components, column vector V_1 is constructed to represent

the failure probability of each component at the beginning of a cascade. $V_1(i) = 1$ denotes component i is part of the initial failures while $V_1(i) = 0$ denotes component i did not fail at the beginning of a cascade. Define $V_1' = e - V_1$ to describe the survival probability of each component where e is an all-ones column vector.

Step 3 Set $G = 2$ (second generation). Given the initial interaction P_0^{current} , the failure probability of each component in the second generation is calculated:

$$V_2(i) = 1 - \prod_{j=1}^{N_1} (1 - V_1(j)P_0^{\text{current}}(i | j)), \quad (3.8)$$

$$V_2'(i) = V_1'(i)(1 - V_2(i)), \quad (3.9)$$

where V_2 and V_2' store the failure and survival probability of each component in the second generation, respectively. N_1 is the total number of initially failed components in the first generation. Following past literature, we assume that component failure probability follows a geometric distribution [78]. If no element in V_2 is larger than a user defined ε , go to Step 6; otherwise, go to Step 4.

Step 4 Set $G = G + 1$. Using the subsequent interaction P_{1+}^{current} , the failure probability of each component in the G th generation is calculated. Note that a system component may be predicted as a potential failure in more than one generation of the propagation tree, e.g., component 3 is predicted to fail in generations 2, 3 and 5 in Fig. 3.3. As a component cannot fail more than once in a cascade, the probability that a component only fails in generation G and not fail in any previous generations is calculated by extracting a subtree from the propagation tree for that component. In this subtree, the component in question only appears in generation G and the potential occurrences of this component in *any* previous generations are removed. For example, for the propagation tree in Fig. 3.3, to calculate the failure probability of component 3 in generation 5, a subtree is generated and shown in Fig. 3.4. It

can be observed that the component 3 only occurs in generation 5 and the potential failures of component 3 in generations 2 and 3 are removed in the subtree. Given the subtree for each component in generation G , its failure probability in generation G is:

$$V_G(i) = 1 - \prod_{j=1}^{N_{G-1}^{\text{sub}}} (1 - V_{G-1}^{\text{sub}}(j) P_{1+}^{\text{current}}(i | j)), \quad (3.10)$$

$$V'_G(i) = V'_{G-1}(i)(1 - V_G(i)), \quad (3.11)$$

where V_G records the component failure probability at generation G , while V'_G stores the survival probability of each component until generation G , i.e., the probability that a component does not fail in the first G generations. $V_{G-1}^{\text{sub}}(j)$ is the failure probability of component j in generation $G - 1$ of the subtree. N_{G-1}^{sub} is the total number of parent nodes of component i in generation $G - 1$ of the constructed subtree.

Step 5 If no element in V_G is larger than ε , go to Step 6; otherwise, return to Step 4.

Step 6 Construction of the propagation tree is stopped.

Once all components stop generating child failures, the propagation tree for a given operating condition is constructed. The failure probability of a component in a cascade can be calculated as,

$$V_{\text{final}}(i) = 1 - \prod_{k=1}^{N_G} (1 - V_k(i)) \quad (3.12)$$

where $V_{\text{final}}(i)$ is the probability of component i fails in this this cascade. N_G is the total number of generations after constructing the propagation tree. It can be observed in the construction steps of the propagation tree that the developed method enables calculating the failure probability of a component in different generations, while the interaction model in [68], its improvement in [69] and the event tree model cannot provide this information.

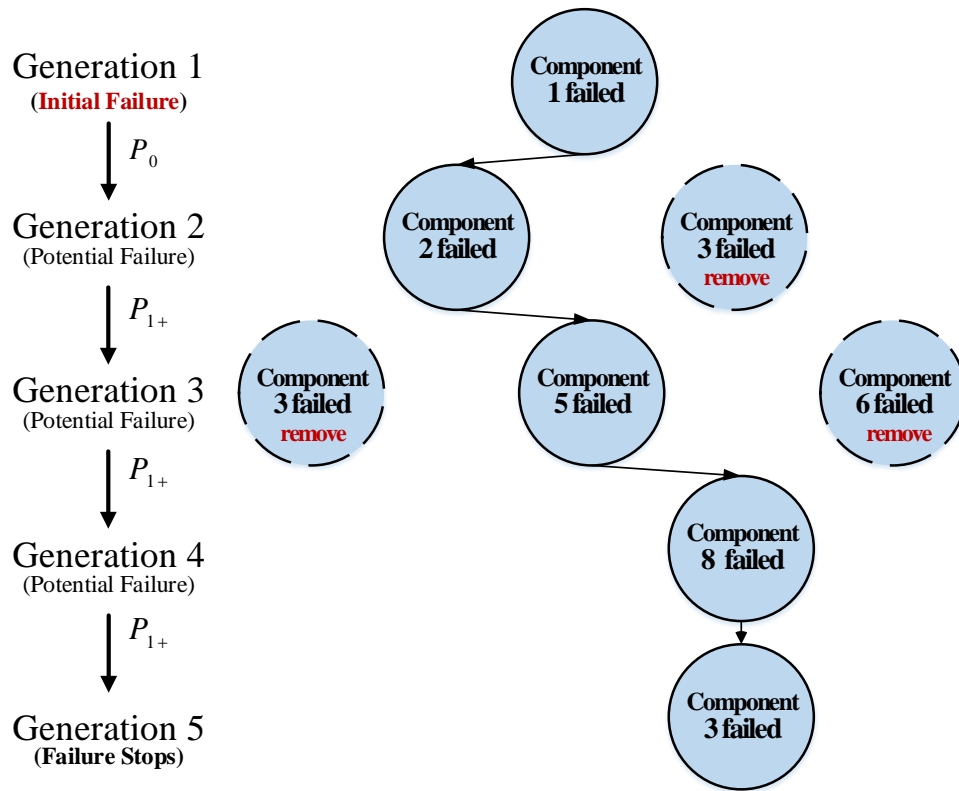


Figure 3.4: An illustration of the subtree of component 3 in generation 5

This is due to the fact that the interaction model ignores the component failure uncertainty, that a component might fail in different generations with different probabilities and only allows components to fail in a single generation. The propagation tree can be used to analyze cascades and identify the most vulnerable areas of a system to cascading failures in different generations. Notice that, when the topology changes due to maintenance, this method can still be applied by updating the initial failure probability of other components that are in service using (3.8).

3.4.4 Cascade Data Sufficiency

So far, the dynamic interaction model assumed that cascade data is sufficiently available for constructing the initial and subsequent interaction matrices and ensuring accurate cascade prediction. Intuitively, more cascade data tends to provide more interaction information between components. However, the question of “how much data is sufficient for

creating interaction matrices?” is yet to be answered.

Recent research studies [68–70] have introduced two criteria to assess the adequacy of cascade data for constructing conventional interaction models. It is shown that with an increase in the number of the cascades in the data, the total number of different interactions between components, i.e., the number of non-zeros in the interaction matrix B increases. However, the total number of different interactions does not significantly change after the number of cascades reaches a threshold. Thus, to obtain most of the interactions between components, the change in the number of non-zeros is used to determine the number of cascades required for building the interaction matrix B . Another lower bound for the number of required cascade data is the number of cascades that can be used to obtain the dominant interactions. By comparing the mismatch of propagation capacity, i.e., the average number of failures in one cascade, between the original cascades and the predicted cascades that are from interaction model, this lower bound of the number of cascades can be determined. The objective of the conventional interaction model is to identify key interactions that largely contribute to the propagation of cascades, while the dynamic interaction model developed in this work is to predict the failure probability of each component. In other words, the entries of the interaction matrix are more important to the dynamic interaction model than the number of non-zeros when deciding on data adequacy for constructing the dynamic interaction model.

The increased number of cascades will change the entries of the dynamic interactions but the change would be limited when the total number of cascades is above a threshold. Here, Frobenius norm is chosen to measure the change in the entries of the interaction matrices and to determine the required number of cascades for building the dynamic interaction model,

$$\begin{aligned}
E^i &= \frac{\|D^i - D^{i-1}\|_F}{N_{\text{nz}}} \\
&= \frac{\sqrt{\sum_{p=1}^{N_{\text{branch}}} \sum_{q=1}^{N_{\text{branch}}} |d_{pq}^i - d_{pq}^{i-1}|^2}}{N_{\text{nz}}}, \quad i = 2, \dots, N_{\text{total}}
\end{aligned} \tag{3.13}$$

where D^i and D^{i-1} are the dynamic interaction matrices, and i is the number of cascades. N_{nz} is the number of non-zero elements in D^i . $\|\cdot\|_F$ denotes the Frobenius norm, E^i indicates the change of the dynamic interaction matrix when the number of cascades is i , d_{pq}^i and d_{pq}^{i-1} are the elements in the p th row and q th column of matrix D^i and D^{i-1} , respectively, N_{branch} is the number of components, and N_{total} is the total number of cascades.

3.5 Simulation Results

The developed dynamic interaction model is evaluated on two test systems. The first test system is the widely used IEEE 30-bus system, with 30 buses and 41 transmission lines, representing a portion of the American Electric Power system [74]. The second system is a synthetic electric grid case, i.e. Illinois 200-bus system, that is statistically and functionally similar to real-world electric grids [79]. This system has a total of 200 buses and 245 lines. The diagrams of these two test systems are given in Fig. 3.5 and Fig. 3.6, respectively. The cascade data is generated from simulations using AC-OPA [48] model. To evaluate the performance of the dynamic interaction model, different performance analysis metrics are used. Also, the developed dynamic interaction model is compared with four baselines: Hines' influence model [66], Qi's interaction model [68], Qi's EM model [69] and dynamic interaction model without differentiating initial and subsequent failures. The goal is to demonstrate the effectiveness of the developed dynamic interaction model in predicting cascades, and highlight its contributions over the existing methodologies.

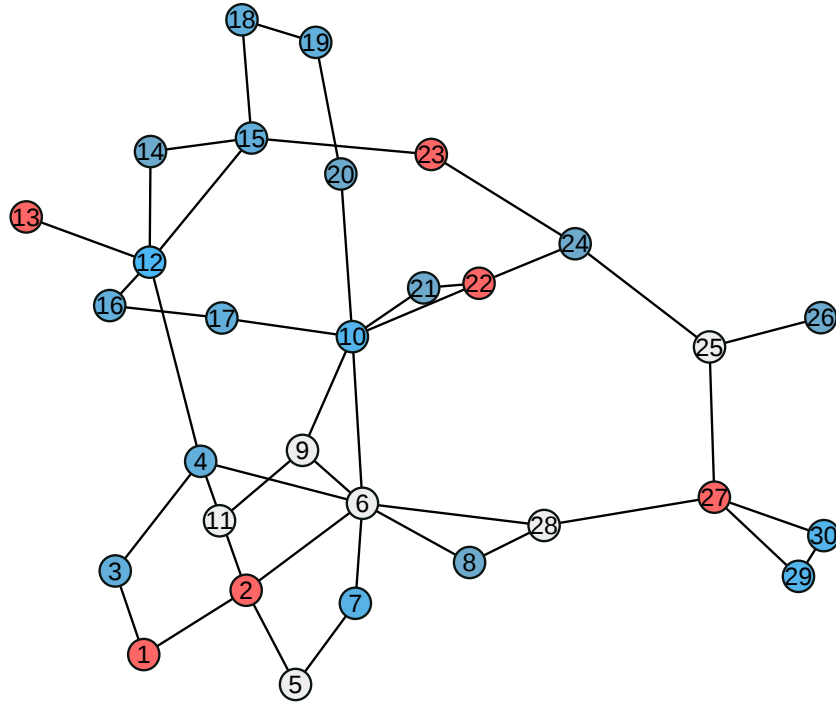


Figure 3.5: Diagram of IEEE 30-bus system that has 30 buses and 41 transmission lines. Red nodes represent generator buses and blue nodes represent load buses.

3.5.1 Cascade Data Preparation

With the aforementioned simulation-based cascade model, a total of 160,000 and 240,000 cascades are generated for the IEEE 30-bus and Illinois 200-bus systems, respectively. By gradually increasing the number of cascades, the changes in the dynamic interaction matrices, in particular, initial interaction and subsequent interactions, for both test cases are illustrated in Fig. 3.7 and Fig. 3.8. Setting the number of bins as two, four different initial interaction matrices that correspond to $P(j|i, r_i \in [0, 0.5), r_j \in [0, 0.5))$, $P(j|i, r_i \in [0, 0.5), r_j \in [0.5, 1.0))$, $P(j|i, r_i \in [0.5, 1.0), r_j \in [0, 0.5))$, $P(j|i, r_i \in [0.5, 1.0), r_j \in [0.5, 1.0))$, and two different subsequent interaction matrices $P(j|i, r_j \in [0, 0.5))$ and $P(j|i, r_j \in [0.5, 1.0))$ are extracted from the cascade data. It can be observed that the change in dynamic interaction matrices for both the initial and subsequent interactions becomes smaller with an increase in the number of cascades, particularly when the number of cascades is small. In other words, more cascade data provides more interaction

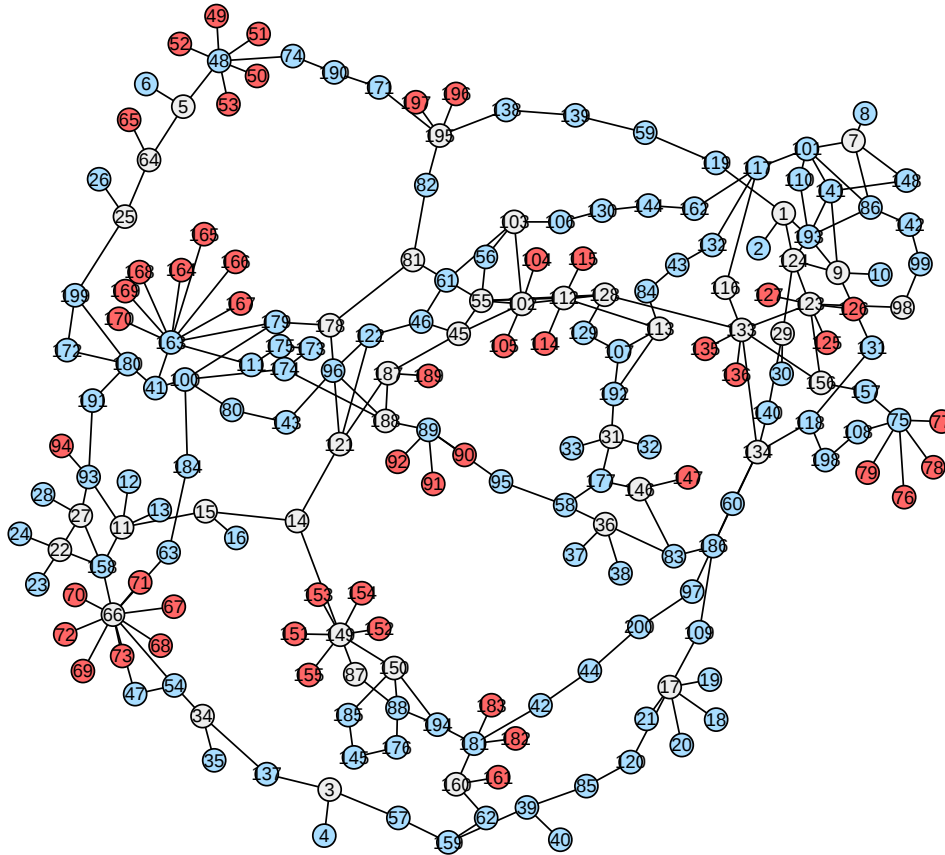
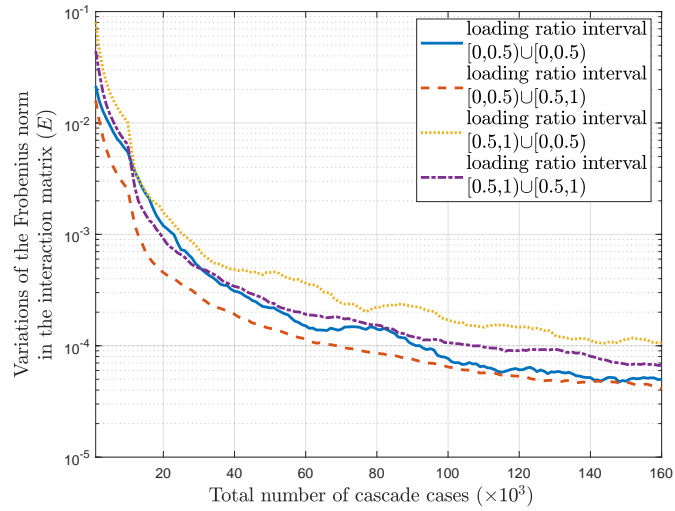
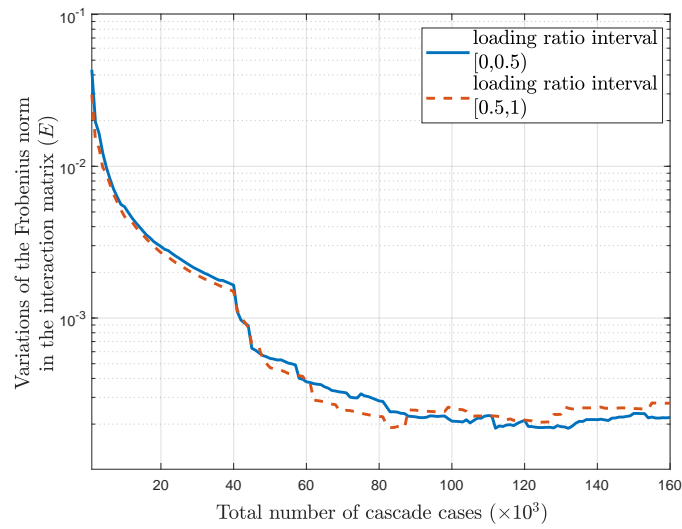


Figure 3.6: Diagram of Illinois 200-bus system that has 200 buses and 245 transmission lines. Red nodes represent generator buses and blue nodes represent load buses.

information. However, when the number of cascades is above a threshold, the Frobenius norm E is close to zero, which means interactions do not change with more cascade data. The change in the Frobenius norm, determines the number of cascade data required for the IEEE 30-bus and Illinois 200-bus systems to be approximately 140,000 and 200,000, respectively.

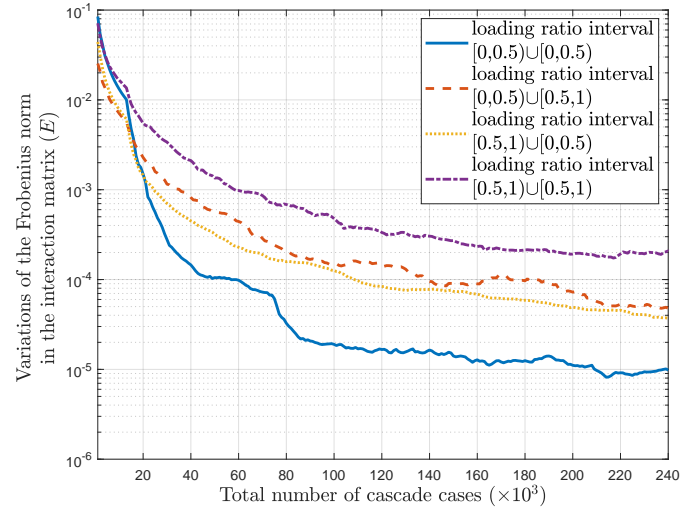


(a) Initial interaction matrices

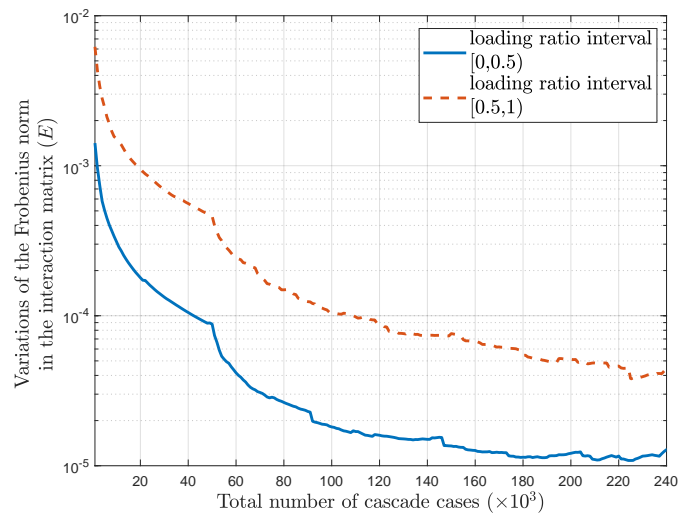


(b) Subsequent interaction matrices

Figure 3.7: Variations of the Frobenius norm calculated for the dynamic interaction matrix for IEEE 30-bus system



(a) Initial interaction matrices



(b) Subsequent interaction matrices

Figure 3.8: Variations of the Frobenius norm calculated for the dynamic interaction matrix for Illinois 200-bus system

3.5.2 Evaluation Metrics

Here we use five metrics to evaluate the prediction accuracy of the dynamic failure model.

1. Jaccard Coefficient

Jaccard coefficient is used to measure the similarity between two sets A_{Truth} and $A_{\text{Predicted}}$ [80]:

$$J(A_{\text{Truth}}, A_{\text{Predicted}}) = \frac{|A_{\text{Truth}} \cap A_{\text{Predicted}}|}{|A_{\text{Truth}} \cup A_{\text{Predicted}}|}, \quad (3.14)$$

where A_{Truth} and $A_{\text{Predicted}}$ are the set of truly failed and predicted failed components in a cascade, respectively. A_{Truth} can be obtained from the original cascade data, while $A_{\text{Predicted}}$ is determined by comparing the total failure probability of each component, i.e., V_{final} in (3.12), with a threshold ε . If the total failure probability of a component is above ε , this component is regarded as a potential failure in a cascade. For temporal failures, A_{Truth} can be obtained from the original data in each generation of a cascade, while $A_{\text{Predicted}}$ is determined by comparing the failure probability of each component in a generation, V_G in (3.10), with a threshold ε .

2. F_1 Score

F_1 score is a weighted average of the *Precision* and *Recall*, which measures the prediction accuracy of the dynamic interaction model [81]. Here, Precision is the fraction of potential failures that indeed fail in a cascade, while Recall is the fraction of failed components that are successfully predicted as impending failures [82].

$$\text{Precision} = \frac{|A_{\text{Truth}} \cap A_{\text{Predicted}}|}{|A_{\text{Predicted}}|}, \quad (3.15)$$

$$\text{Recall} = \frac{|A_{\text{Truth}} \cap A_{\text{Predicted}}|}{|A_{\text{Truth}}|}, \quad (3.16)$$

$$F_1 \text{ score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (3.17)$$

3. Precision@ K (Precision at K)

Precision@ K , inspired by [83], is the fraction of components that have failed among

the top K predicted potential failures:

$$\text{Precision@}K = \frac{|A_{\text{Truth}} \cap A_{\text{Predict, Sort}}^K|}{K}, \quad (3.18)$$

where K is user-defined, and set to the number of components that indeed fail in a cascade. $A_{\text{Predict, Sort}}^K$ is the top K predicted failures in the sorted set of potential failures $A_{\text{Predict, Sort}}$, where potential failures are sorted based on their corresponding failure probabilities.

4. Kendall's Tau (τ)

τ is a correlation coefficient used to measure the association between two ranking methods, where τ is in the range of $[-1, 1]$ [84]. $\tau = 1$ shows a complete agreement, and $\tau = -1$ shows a complete disagreement between the two ranking methods. $\tau = 0$ means the two ranking methods are independent [84]. Given two rankings (orders) $A : a_1, \dots, a_n$ and $B : b_1, \dots, b_n$, consider pair (a_i, b_i) and (a_j, b_j) . If $a_i > a_j$ and $b_i > b_j$ or $a_i < a_j$ and $b_i < b_j$, the pair is concordant; If $a_i > a_j$ and $b_i < b_j$ or $a_i < a_j$ and $b_i > b_j$, the pair is discordant; and If $a_i = a_j$ or $b_i = b_j$, the pair is a tie [85]. τ is formulated as,

$$\tau = \frac{n_c - n_d}{\sqrt{[n_{\text{all}} - \sum_{i=1}^t \frac{t_i(t_i-1)}{2}] \cdot [n_{\text{all}} - \sum_{j=1}^u \frac{u_j(u_j-1)}{2}]}}, \quad (3.19)$$

$$n_{\text{all}} = \frac{n(n-1)}{2},$$

where n_c and n_d are the number of concordant and discordant pairs, respectively. n is the number of elements in the ranking orders A and B , t and u are the number of different group of ties in rank A and B , and t_i and u_j are the number of elements in the i^{th} and j^{th} group of ties in A and B , respectively. For example, given the ranking order A that has seven elements ($n = 7$) as 1 2 2 3 4 4 4, the number of different group of ties are two ($t = 2$) since the rankings 2 and 4 appear more than once, that is, two elements are ranked with the same order 2 ($t_1 = 2$) and three elements are ranked with the order

4 ($t_2 = 3$).

As components in a generation of a cascade either fail or survive, they cannot be ranked; hence, a heuristic ranking method is used for components that have indeed failed at each generation. In this heuristic method, the rank for failed and survived components are set to 1 and 2, respectively. For those components that are predicted to fail, the rank is assigned based on the corresponding failure probabilities. Hence, Kendall's Tau can be used to evaluate the similarity between the predicted and the real failed components in each generation of a cascade.

5. The average precision of the top N generations

The average precision of the top N generations is also assessed in this chapter. Incorrectly predicted failures in one generation of a cascade will lead to wrong predictions in all subsequent generations. As the initial cascade generations are more influential in determining operator actions, the average precision of the top N generations, $Avg_N^{\text{Precision}}$, is used to evaluate the performance of the developed failure prediction methodology.

$$Avg_N^{\text{Precision}} = \frac{\sum_{i=1}^N \text{Precision}_i}{N} \quad (3.20)$$

where Precision_i is the precision for the i th generation. Since most cascades stop within 10 generations, N is set to 10.

3.5.3 Determining Optimal Failure Model Parameters

In addition to the previously collected cascade data for the IEEE 30-bus and the Illinois 200-bus systems, another 50,000 cascades for each test system is generated using the same cascade data generation process to validate the performance of the developed interaction model by using holdout cross validation method [86]. A threshold ε , is used to identify the potential failures in each generation of a cascade. This threshold is varied from zero to

one to find the optimal ε that yields the highest prediction accuracy. The F_1 score of the prediction of total failures with different ε in IEEE 30-bus and Illinois 200-bus systems are given in Fig. 3.9. Here, these values are $\varepsilon = 0.18$ for the IEEE 30-bus system, and $\varepsilon = 0.15$ for the Illinois 200-bus system.

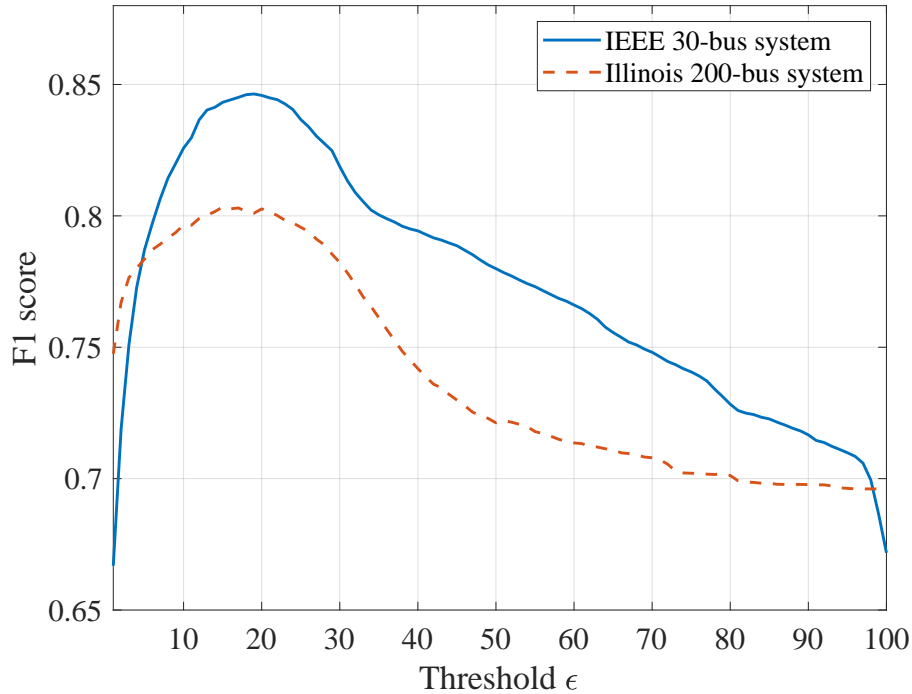


Figure 3.9: F_1 score of prediction accuracy of total failures in IEEE 30-bus and Illinois systems with different $\varepsilon \in [0, 1]$

We found that the prediction accuracy of the developed interaction model increases with the number of bins, b , that are used to discretize dynamic interactions. However, the prediction accuracy did not change as the number of bins increased beyond a certain value. By looking at the prediction accuracy while varying the numbers of bins from 1 to 100, which is given in Fig. 3.10, we have set the number of bins for the IEEE 30-bus and Illinois 200-bus systems to 35 and 40, respectively.

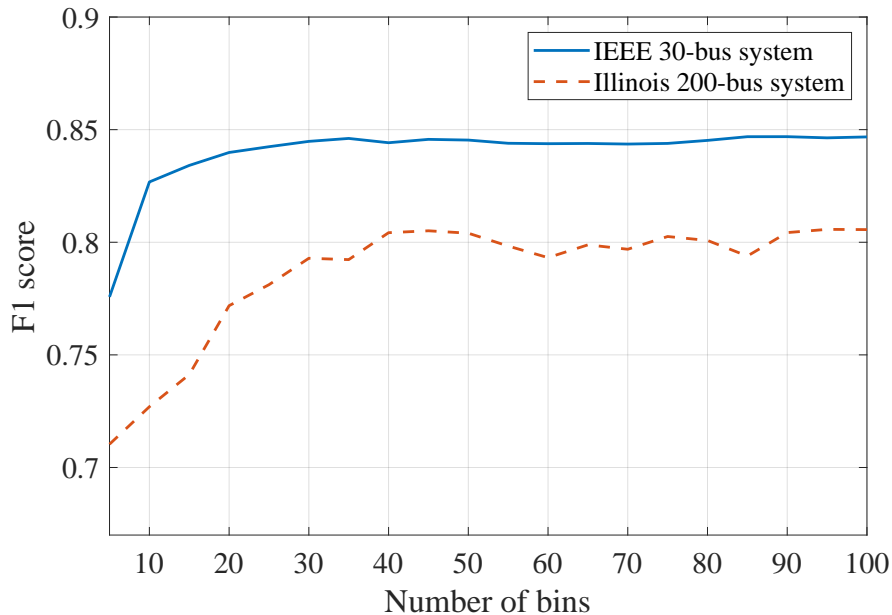


Figure 3.10: F_1 score of prediction accuracy of total failures in IEEE 30-bus and Illinois 200-bus systems with different bins (from 1 to 100)

3.5.4 Performance Analysis

Once the minimum number of required cascade data is determined, the initial and subsequent interactions can be generated and used to predict potential cascades. If a component failure probability in a generation, calculated by the propagation tree, is larger than a threshold ε , this component is regarded as a potential failure in the corresponding generation of a cascade. To evaluate the prediction performance of the developed dynamic interaction model, the five aforementioned metrics are used. The performance of the developed method is analyzed in terms of (1) how accurately all failures in a cascade, referred to as *total failures*, are predicted (not considering which generation each failure happened), and (2) how accurately failures in each generation of a cascade, referred to as *temporal failures*, are predicted.

- **Prediction Evaluation for Total Failures** is performed using three metrics: *Jaccard coefficient*, F_1 score, and *Precision@K*;

- **Prediction Evaluation for Temporal Failures** is performed using Jaccard coefficient, F_1 score, Kendall's Tau, and the average precision of the top N generations to assess failure prediction accuracy in each cascade generation:

The prediction performance based on both total failures and temporal failures for two test systems are provided in Table 3.1 and Table 3.2. It can be observed that, in most cases, the developed dynamic interaction model significantly outperforms the other four baselines in terms of prediction accuracy for both total failures and temporal failures. The performance results of the developed dynamic interaction model with and without separating the interactions also show that dividing the interactions into initial and subsequent interactions has significant improvement in predicting the two different type of failures. Specifically, for the total failures, the Jaccard coefficient of the dynamic interaction model is 0.8159 and 0.7988 for the IEEE 30-bus, and Illinois 200-bus systems, which means the similarity between the predicated failed components and real failed components is 81.59% and 79.88%, respectively. The F_1 score (a weighted average for prediction accuracy) for the two test systems reaches 84.77% and 80.58%, higher than that of the other four baselines. Similarly, the precision@ K values show that 89.72% and 85.26% of the top K predicted failures are correctly predicted, where K is equivalent to the number of components that have indeed failed. Low precision means more lines are incorrectly recognized as risky lines, which would cause unnecessary actions such as relay blocking on those lines. These unnecessary actions are costly and may decrease the system reliability. It can be observed from the performance comparison results that the prediction precision is significantly improved by using the developed dynamic interaction model. In addition to the prediction performance of the total failures, the prediction for the temporal failures also reaches a high accuracy, although the accuracy is slightly decreased. The reason for this reduction in accuracy is that a wrong prediction of a failed component in a generation would cause wrong prediction of failed components in all subsequent generations of a cascade, which can be thought of as a prediction noise. With the propagation of cascades, the effect of

this noise is amplified and results in reduction in the prediction accuracy. For a better illustration of the prediction accuracy in each generation of a cascade, and to decrease the effect of the noise, the average precision of the top N generations for the two test systems are given in Fig. 3.11. We observe that the prediction accuracy decreases with the increase in the number of top N generations. Compared to the other four baselines, the developed interaction model provides a more precise prediction of the failed components in a generation, especially for the top four generations. The values for Jaccard coefficient and Kendall’s Tau also show a high degree of similarity and strong agreement between the predicted outages and the real outages in each generation.

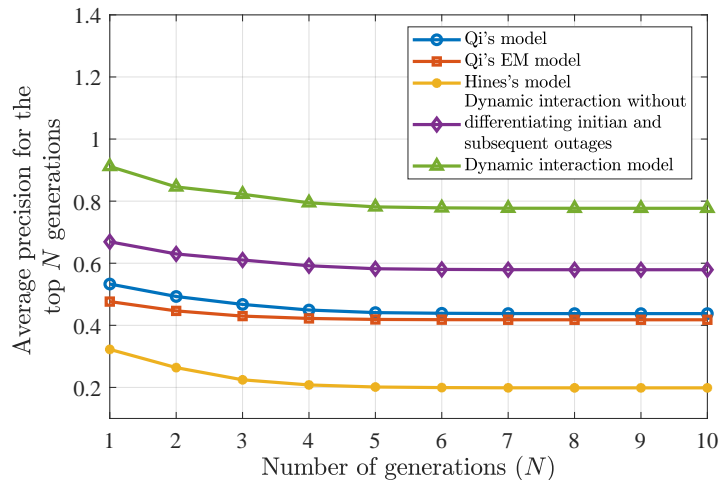
Table 3.1: Performance analysis of the proposed dynamic interaction model for the IEEE 30-bus system

Total Failures					
Metric	Qi’s Model [68]	Qi’s EM Model [69]	Hines’s model [66]	Dynamic interaction model without separating initial and subsequent outages	Developed dynamic interaction model
Jaccard Coefficient	0.5436	0.5598	0.3257	0.7108	0.8159
F_1 score	0.5631	0.5127	0.3344	0.7133	0.8477
Precision@K	0.6409	0.6058	0.3943	0.7713	0.8972
Temporal Failures					
Jaccard Coefficient	0.4203	0.3965	0.1815	0.5631	0.7605
F_1 score	0.4383	0.4058	0.2008	0.5864	0.7866
Kendall’s tau	0.7888	0.7478	0.7976	0.8122	0.8726

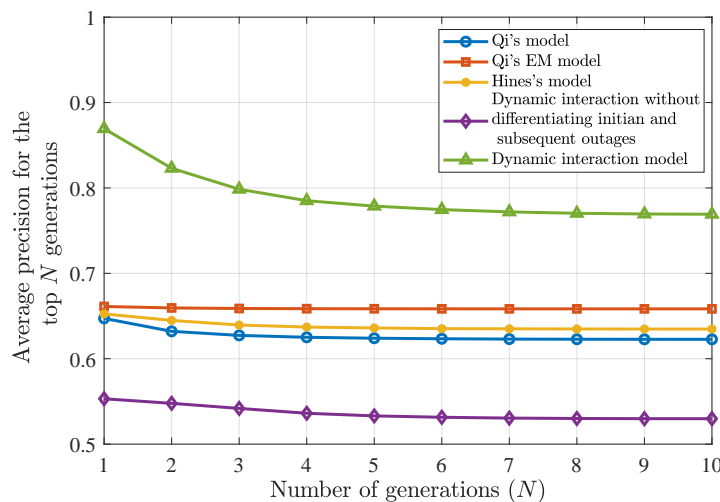
Table 3.2: Performance analysis of the proposed dynamic interaction model for the Illinois 200-bus system

Total Failures					
Metric	Qi's Model [68]	Qi's EM Model [69]	Hines's model [66]	Dynamic interaction model without separating initial and subsequent outages	Developed dynamic interaction model
Jaccard Coefficient	0.6720	0.7126	0.6600	0.5959	0.7988
F_1 score	0.6589	0.6776	0.6649	0.5586	0.8058
Precision@K	0.7059	0.7690	0.7448	0.6274	0.8526
Temporal Failures					
Jaccard Coefficient	0.6196	0.6577	0.6316	0.5219	0.7585
F_1 score	0.6220	0.6580	0.6358	0.5273	0.7711
Kendall's tau	0.8216	0.8360	0.8504	0.8575	0.8692

The computation time of the dynamic interaction model is also reported here to assess its suitability in near real-time. The time required to online analysis of 1,000 cascades for IEEE 30-bus and Illinois 200-bus systems using the dynamic interaction model are 8.1043 s and 250.2494 s, respectively, on a computer with an i7-7700 CPU, 4.2GHz core. Compared to the time it takes to analyze the same number of cascades using AC cascading failure model, which is 300.7678 s for the IEEE 30-bus system and 1371.4187 s for the Illinois 200-bus system, the time efficiency of the developed model is a clear improvement. Although the OPF approach in AC cascading failure model is fast if pre-calculated solutions of some specific operation conditions are stored, it is impractical to calculate and store solutions for all possible system operating conditions due to the numerous uncertainties present in today's power grids, particularly from renewable energy resources. It can be observed that the computation time would be increased with the size of the system. In terms of the type of the system, a well-connected network will resist more cascade propagation, as it is more robust. Hence, the number of generations in a cascade in a well-connected power system would be less than other loosely connected systems. As a



(a) IEEE 30-bus system



(b) Illinois 200-bus system

Figure 3.11: Average precision with top N number of generations

consequence, the former would have shorter computation time using propagation tree due to the less number of cascade generations that need to be estimated. Furthermore, in order to make the developed dynamic interaction model more suitable for online applications, several strategies are proposed here to speed up the computation time: 1) Instead of evaluating all $N-1-1$ and $N-2$ contingencies, it is suggested to apply the dynamic interaction on the selected credible contingencies that can be obtained from various contingency selection methods [87–91]; 2) Only forecast the propagation of the first K_{predict} generations of a cascade as the initial progress of the failure propagation is more important to system

operators to identify potential failures; 3) High performance computing can further reduce the computation time.

3.6 Discussion

Analyzing the performance of different statistical cascading failure models has shown that the developed dynamic interaction model provides more accurate prediction of the total and temporal failures in each generation of a cascade. With the dynamic interaction model, one is able to identify components that have high failure probability in the subsequent failures, particularly at the beginning of a cascade, where the progress of the failures is slow [92]. Therefore, operators can take targeted control actions that reduce the loading ratio of the identified potential components or disable the corresponding relays to stop the tripping of these components. The examples of cascade prediction for the IEEE 30-bus and Illinois 200-bus systems are illustrated in Fig. 3.12 and Fig. 3.13. The real sequence of the two cascades that are obtained from the AC-OPA model is given in Table 3.3.

Table 3.3: Example of two real sequence of failures in two test systems

	IEEE 30-bus system	Illinois 200-bus system
Generation	Failed components	
1	12, 36	41, 144, 208
2	10, 29, 30, 31, 33, 35	9, 10, 83, 112, 229, 230
3	16, 28, 34, 40, 41	8

It can be observed in Fig. 3.12 that, given the initial failures in generation one, i.e. lines 12 and 36 are out in IEEE 30-bus system, the obtained failure probabilities of lines 10, 29, 30, 31, 33 and 35 from the dynamic interaction model are close to one, which are in accordance with the real failed components in generation 2 in Table 3.3. It is also seen that the real failed components, i.e. components 16, 28, 34, 40 and 41, have high failure

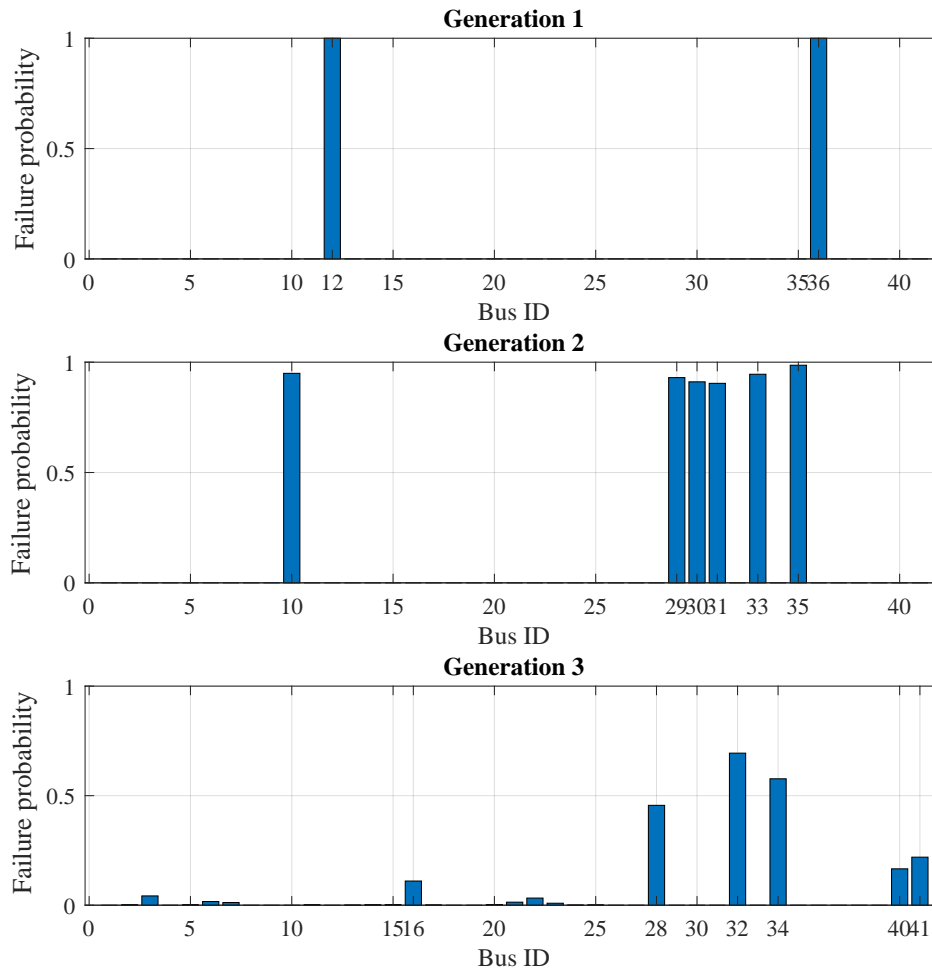


Figure 3.12: Potential failures in each generation of a cascade in IEEE 30-bus system.

probability in generation three in Fig. 3.12. However, line 32 has high failure probability in the generation 3 in Fig. 3.12 while it does not fail in the real case. Further investigation shows that the power flow of line 32 is close to its limit. Comparing the real failure sequence in Illinois 200-bus system with Fig. 3.13, the same observation is made: the dynamic interaction model effectively identifies components that are prone to failure in each generation. The only exception is the wrongly predicted (False positive) failure of line 207 in the second generation. Although line 207 does not get overloaded, it is found that the power flow of line 207 is close to its thermal rating. It can be observed from the illustration results that the dynamic interaction model provides useful insights into the potential failures in each generation of a cascade, that in turn can be used by the system

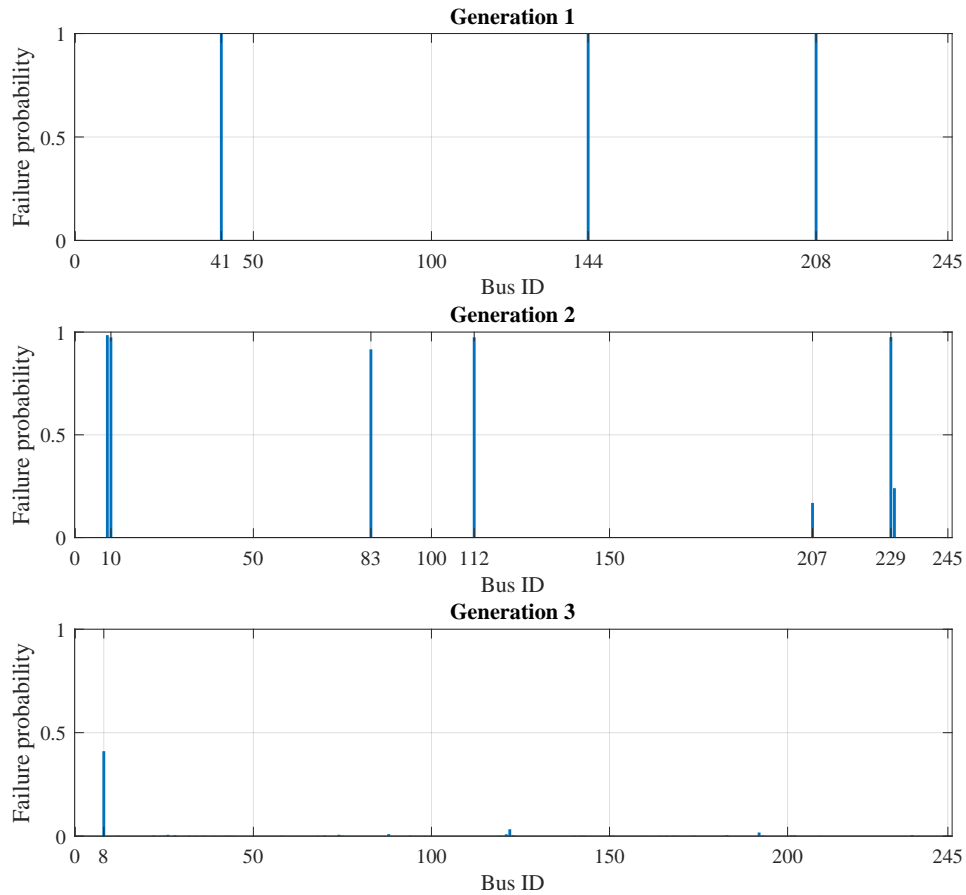


Figure 3.13: Potential failures in each generation of a cascade in Illinois 200-bus system.

operators to take effective actions that prevent the propagation of cascades, in a timely manner.

3.7 Conclusion

This chapter presents the *dynamic cascading failure model*, which facilitates prediction of impending cascading failures under a specific power system operating condition. The conventional statistical cascading failure models only capture the general propagation patterns of cascades. On the contrary, the dynamic failure model incorporates system operating conditions into the component failure interactions using a Bayesian approach, and thus enables identifying component failure interactions that correspond to the latest available line power flows. By analyzing simulated or historical cascade data, the interactions be-

tween system components are estimated offline, and used in online operations to calculate the failure probability of each system component using the introduced *propagation tree* model. The prediction efficiency of the developed dynamic failure model is evaluated by various metrics, showing a significant improvement over four other state-of-the-art methods in predicting total and temporal failures of a cascade. When an unexpected power system failure occurs, the system operators can promptly identify the most probable component failures with the developed failure model. Hence, proper actions can be taken to prevent further loss of system components.

Chapter 4

Impact Prediction of Power System

Events in Transient State

4.1 Introduction

In our research, we have analyzed the PMU-based means for assessing the impact of failures in terms of steady-state in Chapter 5. With this tool, the impact of system failures can be promptly minimized, and thus prevent the propagation of cascading failures. However, it is not sufficient for improving system reliability, as the dynamics of the power system have not been taken care of. When a three-phase fault occurs on a transmission line, all generators' rotor angles will swing together. One example of the dynamics of relative generator rotor angles and bus voltage magnitudes is depicted in Fig. 4.1. As seen, large excursions occur in the trajectories of the generator rotor angles and bus voltage magnitudes when the described fault occurs. This example has a stable transient as all generators can remain synchronized. If generators cannot maintain synchronism, referred to as unstable transient, the grids may collapse and require immediate control actions, such as controlled islanding, to split the unsynchronized generators. The example of the unstable transient is illustrated in Fig. 4.2. It can be observed that, after a system failure, one of the generators cannot keep synchronism with others.

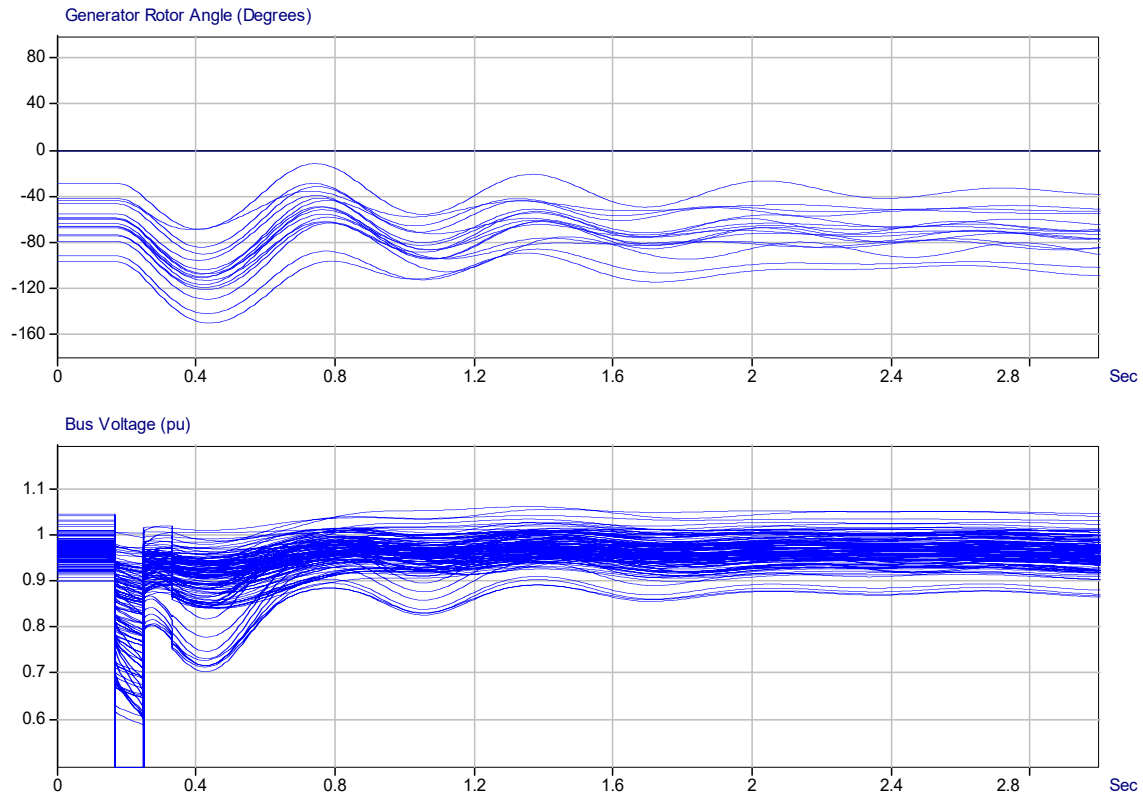


Figure 4.1: The trajectories of the generator rotor angles (19 generators) and voltage magnitudes (118 buses) in IEEE 118-bus system. A three-phase short circuit has occurred on line 102 at 0.167s and is cleared after 5 cycles (0.083 s).

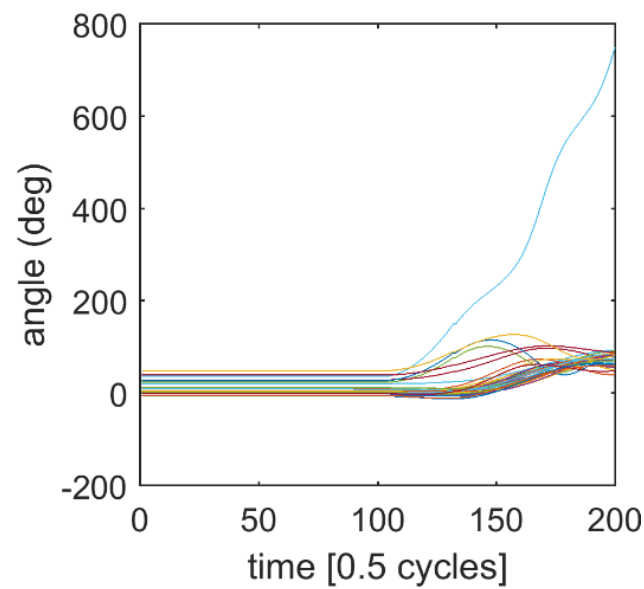


Figure 4.2: Illustration of the unstable transient after a fault.

In order to maintain stable transient, it is crucial to equip the grid operators with a fast, on-line transient stability assessment (TSA) tool to accurately determine the system dynamic performance, at the earliest stage of the event. Thus, system operators can be granted with adequate time to take proper mitigative actions, such as intentional islanding or controlled load shedding.

As PMUs enables generating accurate measurements with high resolutions, the dynamics of the system can be better observed. Here we develop a new PMU-based data-driven TSA approach to predict the transient stability of a system while requiring much less transient data than the conventional methods. The data reduction is enabled by learning the dynamic behaviors of the historical transient data using generative and adversarial networks (GAN). This knowledge is used online to predict the voltage time series data after a transient event. A classifier embedded in the generative network deploys the predicted post-contingency data to determine the stability of the system following a fault. The developed GAN-based TSA approach preserves the spatial and temporal correlations that exist in multivariate PMU time series data. Hence, in comparison with the state-of-the-art TSA methods, it achieves a higher assessment accuracy using only one sample of the measured data and a shorter response time.

4.2 Background

The conventional TSA methodologies belong to one of three categories: 1) time-domain simulations, 2) direct methods, and 3) trajectory-based approaches. Time-domain methods solve a set of high-dimensional and nonlinear differential algebraic equations to assess the transient stability [93, 94]. These methods assume complete knowledge of the system model parameters and the operating conditions. However, their high computational complexity makes them impractical for near real-time applications. To enable faster transient stability assessment, direct methods, e.g., Lyapunov method [95–97] and extended equal

area criterion [98, 99], have been developed. The direct methods simplify TSA by energy functions that evaluate the dynamic performance of the system in a shorter time [100, 101]. Yet, because of the need to simplify the dynamic models, these methods do not scale well to large power systems. The trajectory-based methods, such as Lyapunov exponents [102] and apparent impedance [103] methods, are relatively fast. However, at least a few cycles of post-fault transient data are needed for a precise stability assessment.

With the increased availability of high-resolution Phasor Measurement Unit (PMU) data, new TSA approaches have emerged where physical system constraints and system data inform each other for a more accurate TSA. Hence, without constructing a complete dynamic grid model, the dynamic stability status of power grids can be determined. Machine learning methods such as decision trees [103–105], core vector machine [106], support vector machine (SVM) [107], and extreme learning machine [108, 109] have previously been deployed for TSA using system measurements, formulating TSA as a two-class (stable and unstable) classification problem. These data-driven methods require a conversion from raw Phasor Measurement Unit (PMU) data to selected features, which may result in the loss of critical dynamic information and adversely affect the classification accuracy. Deep learning methods, such as convolutional neural network (CNN) [110, 111], long short-term memory (LSTM) [112, 113], gated recurrent unit (GRU) [114], and the stacked denoising autoencoder (SDAE) [115] have been shown to address the problem of information loss by directly utilizing raw PMU data. However, to guarantee the assessment accuracy, longer durations of post-contingency time series data are needed, which will result in TSA delays. Longer delays are not desired for online TSA since the window of opportunity for taking an effective corrective action may be missed.

4.3 Hierarchical GAN-based Transient Stability Assessment

As explained in the Section 4.2, the existing TSA methods are not adequate for online prediction of the transient status. To address the aforementioned gaps in data-driven TSA methods, a neural network model, i.e., generative and adversarial network (GAN) model, is extended in this section to *predict* the post-contingency PMU measurements, that will be used for classification. The novelty of the developed TSA approach stems from the ability to rapidly and accurately determine transient stability, while only utilizing one sample of the measured post-contingency PMU data. This is achieved by developing a hierarchical structure, shown in Fig. 4.3, for the refined generative adversarial networks. Several GAN models are stacked together to construct HGAN. The input to the hierarchical generative adversarial network (HGAN) is the measured post-contingency PMU data. The lowest level GAN utilizes the measured PMU data to predict the measurements for the next sampling time. The subsequent predictions are used by higher level GANs to further predict the PMU time series data. With the proposed structure, HGAN requires only a single sample of the measured data to predict the post-contingency time series transient data. In addition to the predicted sequence of the transient data, a binary classifier is embedded in the generative network of each sub-GAN. Each sub-GAN performs TSA individually, based on the measured one-sample PMU data as well as the predicted subsequent time series data from the lower-level sub-GANs. The final TSA is achieved by combining the assessment results from all sub-GANs, with a goal to reduce the prediction error.

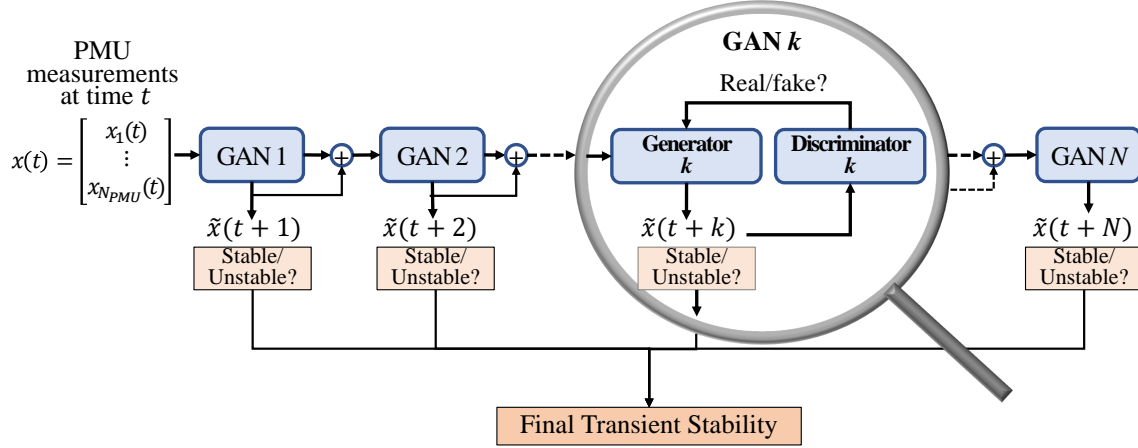


Figure 4.3: The structure of the developed HGAN-based TSA

4.3.1 Transient Stability Assessment

During a power system disturbance, such as a transmission line fault, large swings appear in generator rotor angles, bus voltages, or line currents [115]. To ensure a stable grid, all generators should maintain synchronism after the clearance of the faults. If the synchronism is lost, prompt mitigations such as controlled islanding need to be taken to prevent further failures. This remarks the need for an online TSA approach that can promptly and precisely identify the transient status at the earliest stages of a fault clearance.

Assume a power system has n generators and the loads are constant impedances, the transient of the generator i can be modeled as [116, 117],

$$\dot{\omega}_i = \frac{P_i^m - P_i^e - D_i \omega_i}{M_i} \quad (4.1.1)$$

$$\dot{\delta}_i = \omega_i \quad (4.1.2)$$

$$P_i^e = E_i^2 G_{ii} + \sum_{j \neq i}^{n+1} C_{ij} \sin(\delta_{ij}) + \sum_{j \neq i}^{n+1} D_{ij} \sin(\delta_{ij}) \quad (4.1.3)$$

$$C_{ij} = E_i E_j B_{ij} \quad (4.1.4)$$

$$D_{ij} = E_i E_j G_{ij} \quad (4.1.5)$$

$$\delta_{ij} = \delta_i - \delta_j. \quad (4.1.6)$$

Here, a node $n + 1$ is modeled as the infinite bus such that $E_{n+1} = 1$ and $\delta_{n+1} = 0$. From (4.1.1) and (4.1.2), it can be observed that the rotor speed, electrical and mechanical powers are the functions of rotor angle $f(\delta_i)$. Since the rotor angle is typically used to determine the transient stability, analyzing these three terms, i.e., P^m , P^e , and ω , can help indirectly study the transient. When a severe disturbance occurs in power grids, the electrical power will immediately change, while the mechanical power and rotor speed will respond later due to the inertial and governor in the prime mover. The different response time can lead to imbalanced electrical and mechanical power, affecting the entire rotating system of the grid, and may cause unstable transient. As the imbalanced power is mainly due to the sudden change of the electrical power P^e , the change of the electrical power after the occurrence of the disturbance can be used to learn the transient stability. Furthermore, as observed in (4.1.3) that the electrical power P^e is the function of the voltage, the trajectory of the voltage can thus be used to study the transient stability.

Fig. 4.1 depicts the post-fault dynamics of the generator rotor angles and bus voltages in the IEEE 118-bus system. A three-phase short circuit is simulated on line 102 at time instant 0.167s, which is cleared after 5 cycles, and Generator 1 is the reference machine. As seen, large excursions occur in the trajectories of the generator rotor angles and bus voltage magnitudes when the described fault occurs. To classify the system transient stability status, a commonly used stability index based on rotor angles is used [118]:

$$\eta = \frac{360^\circ - \Delta\sigma_{max}}{360^\circ + \Delta\sigma_{max}} \quad (4.2)$$

where $\Delta\sigma_{max}$ is the maximum post-fault angle difference between any two generators.

The stability is defined as,

$$\phi = \begin{cases} \text{stable}; & \eta > 0 \\ \text{unstable}; & \eta \leq 0. \end{cases} \quad (4.3)$$

In the transient event depicted in Fig. 4.1, $\eta = 41.3$. Therefore, the IEEE 118-bus system is considered stable after clearing the three-phase fault on line 102. However, using (4.2) for determining the transient status requires the estimation of $\Delta\sigma_{max}$ over a relatively long time (a few seconds), which is not desired for online TSA. Hence, the objective of the developed TSA approach, illustrated in Fig. 4.4a, is to acquire the transient status at the earliest stages of a contingency. As illustrated in Fig. 4.4b, unlike the other machine learning-based TSA methodologies that require a large volume of post-contingency PMU data, the developed method only requires the first sample of the PMU data after the clearance of the faults. Thus, more time can be granted for finding the optimal corrective actions. In this work, the PMU voltage data are selected for TSA. As mentioned in [119], the generator rotor angle can not be measured directly by PMUs. Since the focus of this section is to determine the system transient stability from PMU data, PMU voltage magnitudes are used. Various data-driven TSA methods have used voltage data to study the transients [112, 119–122]. The studies in [121] and [122] demonstrate that voltage magnitude-based transient algorithms are faster, simpler, and could eliminate the errors and extra pre-processing associated with the calculation of an angle reference. Additionally, the dynamic equation (4.1) has shown that the bus voltage data has abundant information of the transient. It is also shown in Section 4.4.2 that only utilizing PMU voltage magnitude data is sufficient to achieve a high assessment accuracy. However, other PMU measurements that contain the system response, such as bus voltage angles and line current phasors, can also be deployed by the developed HGAN-based TSA method. Additionally, to mimic real power grids, only some buses are equipped with PMUs (e.g., 20 buses in the IEEE 118-bus system). The PMUs are added in a manner to ensure full system observability.

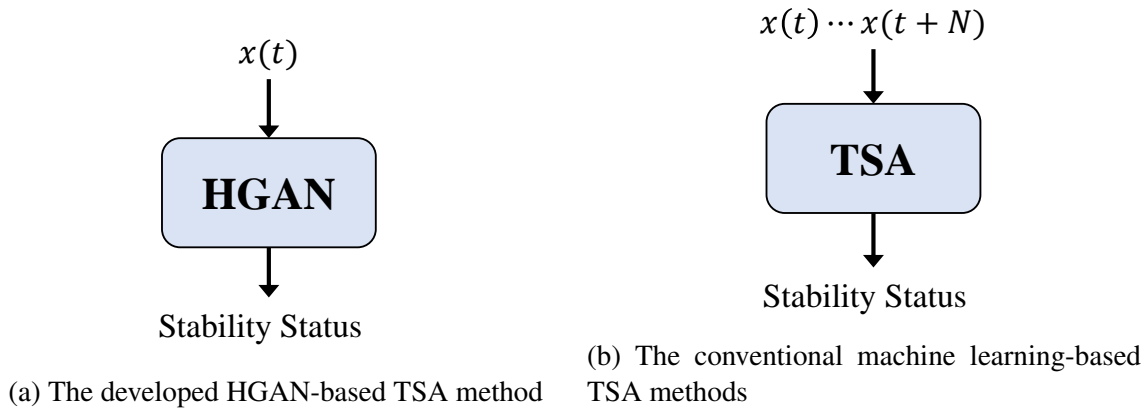


Figure 4.4: The input and output of the developed HGAN model and the conventional machine learning-based TSA methods. $x(t)$ is the first sampled post-contingency PMU data. The number of PMU samples is N .

4.3.2 The Generative Adversarial Network

GAN is an unsupervised learning approach that has been used for generating synthetic images, music, and other time-series data [123–127]. GAN learns the distribution of a real dataset and generates synthetic data that are close to real data [128]. GAN-based methods have been previously developed to recover missing PMU data [129] and generate synthetic PMU datasets [130–132].

GAN consists of two deep neural networks: generator G and discriminator D . The generator generates synthetic data while retaining the statistical features of the real data. The discriminator differentiates the real data from the synthetic data. The two networks are trained iteratively until a Nash equilibrium is achieved and the generated synthetic data and real data cannot be distinguished [129].

For a random Gaussian noise space \mathbf{Z} , the generator maps the input noise z drawn from \mathbf{Z} to a synthetic dataset \hat{x} as,

$$\mathbf{G}(z|\theta_g) : z \rightarrow \hat{x} \quad (4.4)$$

where θ_g is the neural network parameters of G . The objective of the generator is to map a random noise to synthetic data, such that the distribution of the synthetic data $P(\hat{x})$ is

close to the real data $P(x)$. The discriminator D estimates the probability, $D(\cdot|\theta_d)$, that the input data is real rather than synthetic, where θ_d the is the neural network parameters of D . The discriminator aims to maximize the difference between the real and synthetic data so that they could be distinguished,

$$\begin{aligned} \text{Max}_D V(G, D) = & \mathbb{E}_{x \sim P(x)}[\log D(x)] \\ & + \mathbb{E}_{z \sim P(z)}[\log(1 - D(\hat{x}))] \end{aligned} \quad (4.5)$$

where $V(\cdot)$ is the loss function. The first term $\mathbb{E}_{x \sim P(x)}[\log D(x)]$ is the probability that the discriminator classifies the real data as real. The second term is the probability that the synthetic data \hat{x} generated from G is classified as fake by D . The objective of G is to minimize the difference between the real and synthetic data,

$$\text{Min}_G V(G, D) = \mathbb{E}_{z \sim P(z)}[\log(1 - D(\hat{x}))]. \quad (4.6)$$

As the generator and discriminator are trained together, the objective of GAN is to address a minimax problem:

$$\begin{aligned} \text{MinMax}_{G, D} V(G, D) = & \mathbb{E}_{x \sim P(x)}[\log D(x)] \\ & + \mathbb{E}_{z \sim P(z)}[\log(1 - D(\hat{x}))]. \end{aligned} \quad (4.7)$$

The GAN training algorithm is described in Algorithm 1. In each training episode, by leveraging the stochastic gradient descent algorithm, the generator and discriminator are updated, such that the synthetic data are closer to the real data.

Algorithm 1 The GAN training steps

Initialize the neural network parameters of the generator and discriminator, i.e., θ_g and θ_d respectively **for** $ep = 1, \dots, ep_{max}$ **do**

Obtain a minibatch of the random noise z from $P(z)$, and another minibatch of the real data x from $P(x)$ Generate the synthetic data $\hat{x} = G(z|\theta_g)$ Feed the generated synthetic data \hat{x} and the real dataset x into the discriminator D , and obtain the estimated probability $D(x|\theta_d)$ and $D(\hat{x}|\theta_d)$ Update G and D by descending their gradients:

$$\nabla_{\theta_g}(\mathbb{E}_{z \sim P(z)}[\log(1 - D(\hat{x}))])$$

$$\nabla_{\theta_d}(\mathbb{E}_{x \sim P(x)}[\log D(x)] + \mathbb{E}_{z \sim P(z)}[\log(1 - D(\hat{x}))])$$

end

4.3.3 Gated Recurrent Units for Feature Extraction from PMU Time Series Data

When predicting the future dynamic response in a power system, it is critical to preserve the spatial and temporal features of the time series data. Gated Recurrent Units (GRUs), a variant of RNN, enable this by learning the spatial and temporal features of time series data [133].

Equipped with directed cycles in network connections, RNN retains information from past data, so that it can be combined with the present data to more accurately predict the future sequence of data. However, RNN suffers from the vanishing or exploding gradient problem during training [134]. If the RNN is not designed properly, the gradient may be exponentially updated towards zero, referred to as a vanishing gradient, or exponentially diverge to infinity, referred to as an exploding gradient. Either of these problems challenges the proper RNN training. To resolve this problem, GRU is equipped with a specific mechanism to determine the volume of past data, to be added to the output. By doing so,

the problem of vanishing or exploding gradient is eliminated. As illustrated in Fig. 4.5, GRU consists of three elements: the reset gate, the update gate, and the current memory unit [133]. The update gate determines the volume of past data that are used for predicting the future data sequence, while the reset gate determines the volume of the stored past information to be forgotten. The current memory unit uses the output from the reset gate to store the relevant information from the past data. The output of GRU is the sum of the output from the three gates. The three gates and the final output can be described as,

$$r_t = \sigma(W^r x_t + U^r h_{t-1}) \quad (4.8.1)$$

$$z_t = \sigma(W^z x_t + U^z h_{t-1}) \quad (4.8.2)$$

$$h'_t = \tanh(W x_t + r_t \odot U h_{t-1}) \quad (4.8.3)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t. \quad (4.8.4)$$

In (4.8), W and U are the weights of the GRU network, and \odot is the Hadamard (element-wise) product. With this specific GRU structure, the transient sequence can be predicted. The premise of this method is to use only one sample of the measured PMU data. This critical advantage is enabled by learning the spatial and temporal features of the transient data. The added discriminator to the GRU, enables further capturing the distribution of the transient data.

4.3.4 The Developed HGAN-based TSA Method

The developed HGAN-based TSA, shown in Fig. 4.3, aims to predict the post-fault transients and assess the system's transient stability. The GAN structure is also improved by learning the spatial and temporal features of the multivariate PMU time series data. As deploying one type of PMU data is sufficient to assess the transient stability, the PMU voltage magnitude data are utilized for TSA. However, other PMU measurements could also be used. Additionally, the developed HGAN adjusts the conventional GAN model

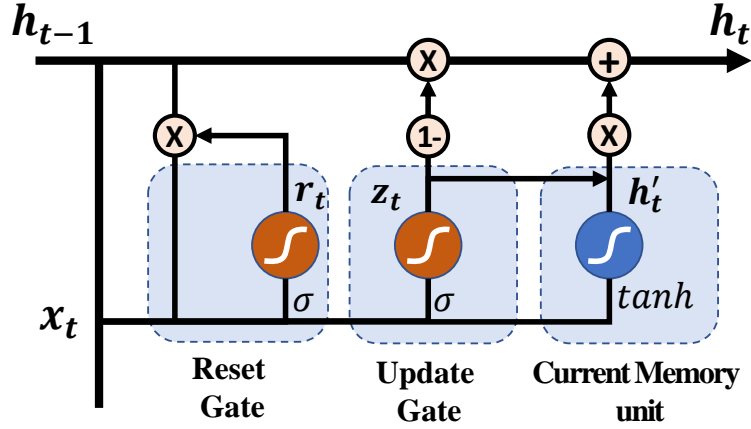


Figure 4.5: The structure of GRU

to predict the PMU sequence data. Conventional GAN takes random noises to produce synthetic data, while the refined GAN takes the real measurements as input to predict the transient sequence. Here, the measured post-fault PMU voltage magnitudes are the inputs to the GAN model, i.e., $x(t)$. The data for the next time instant, i.e., $\hat{x}(t+1)$ are predicted by the aforementioned GAN structure. However, the measured data $x(t)$ and the predicted data $\hat{x}(t+1)$ may not have sufficient information to determine the transient status. To address this deficiency, a recursive strategy for multi-step prediction is developed. The developed model uses the predictions from prior time steps to forecast the values in future time steps. The developed strategy is to stack multiple GANs to predict a longer sequence of future data. As a result, a higher transient assessment accuracy can be achieved. With N stacked GANs, the measured PMU post-fault voltage data is fed into the lowest level GAN, i.e., *GAN 1* to predict the data for the next time instant $\hat{x}(t+1)$. The predicted data and the measured data $x(t)$ and fed into *GAN 2* for subsequent predictions. Therefore, the HGAN-based TSA with N GANs generates $N+1$ voltage time series data, $\{x(t), \hat{x}(t+1), \dots, \hat{x}(t+N)\}$.

A binary classifier is embedded in each generator of the stacked GANs for TSA classification. In each *GAN k*, the embedded classifier utilizes the input sequence data $\{x(t), \hat{x}(t+$

$1), \dots, \hat{x}(t+k-1)\}$ to determine the TSA classification, i.e., stable or unstable. The classification results from all sub-GANs are then combined by using an average-based ensemble strategy to determine the final transient stability assessment. The generator has three objectives: 1) learning the distribution of the real measurements; 2) using the learned distribution to predict the next sequence of the time series data; 3) assessing the transient stability status. Based on these objectives, the generator loss function in *GAN* k is formulated as,

$$\begin{aligned} \mathbf{L}_{G_k} = & \log(1 - D(\hat{x}(t+k))) + (\hat{x}(t+k) - x(t+k))^2 \\ & + y \log(p(y)) + (1 - y) \log(1 - p(y)) \end{aligned} \quad (4.9)$$

where y is the binary label of the transient data. Here, a zero label is assigned to unstable transients, and one to stable transients. The probability that the data is labeled as stable is defined by $p(y)$. In formulating the loss function in (4.9), the square of the error, $(\hat{x}(t+k) - x(t+k))^2$, is used to minimize the error of the predicted data $\hat{x}(t+k)$. The cross entropy loss $y \log(p(y)) + (1 - y) \log(1 - p(y))$ is included so that the generator correctly classifies the system stability status. The structure of the discriminator D in the developed model is consistent to the conventional GAN, where the difference between the real and predicted data will be maximized. In addition, the GRU structure is used to construct generative and adversarial neural networks to better learn the spatial and temporal features of the time series data.

The detailed structure of *GAN* k ($k \in \{0, 1, \dots, N\}$) is illustrated in Fig. 4.6. The input to *GAN* k is the measured and predicted PMU voltage data $X = \{x(t), \hat{x}(t+1), \dots, \hat{x}(t+k-1)\}$, $X \in \mathbf{R}^{k \times N_{PMU}}$, where N_{PMU} is the total number of PMUs deployed in the system. The input X is first fed into the GRU cell in the generator to learn the spatial and temporal features of the time series data. To improve the learning efficiency, the fully connected layer is added after the GRU cell. As the generator has two different outputs,

i.e., the predicted label and subsequent data, two parallel fully connected layers are used to separately perform these two tasks. To generate the predicted label y and the predicted data $\hat{x}(t+k)$, the *SoftMax* and *Sigmoid* activation functions are deployed to interpret the fully connected layer output. GRU is also leveraged in discriminator to construct the neural networks. The output of the GRU in the discriminator is sent to the fully connected layer, followed by a linear activation function that converts the output to the estimated probability of the input data $D(\cdot|\theta_d)$. The probability is backpropagated to the generator to update the parameters of the generative network.

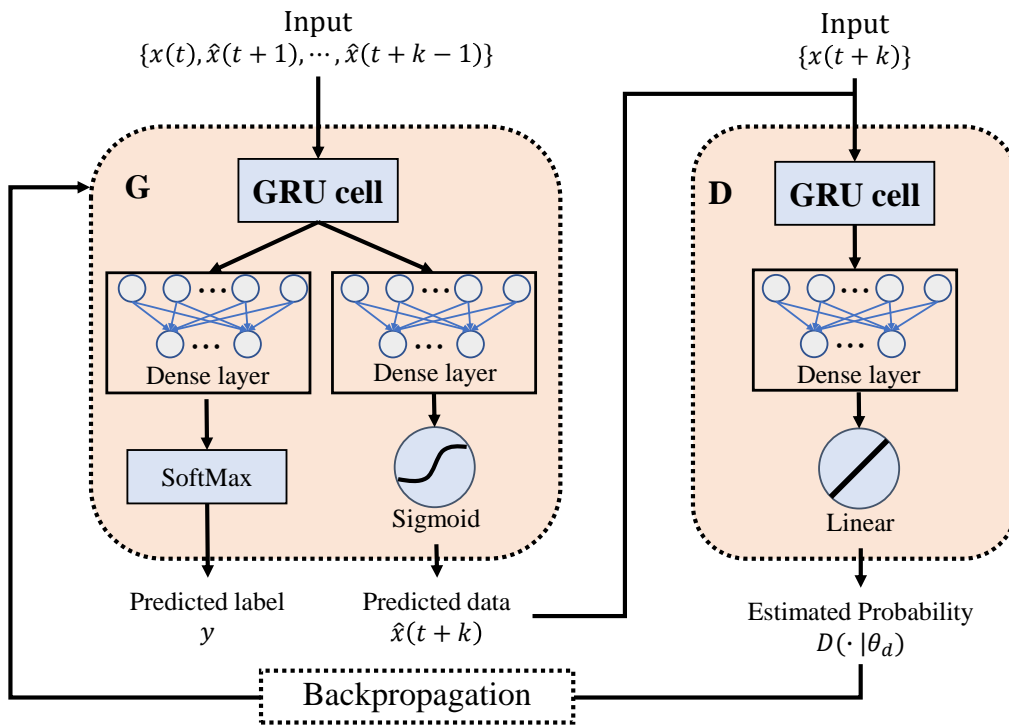


Figure 4.6: The structure of GAN k in the HGAN-based TSA

The prediction accuracy of the upper level GAN depends largely on the accuracy of the lower GANs. A small prediction error from a lower level GAN will propagate to the higher levels and eventually lead to a large prediction error. Hence, in this section, the GANs are trained sequentially. For instance, when training GAN k , the network parameters of the well-trained lower GANs i , ($i \in \{0, \dots, k-1\}$), are kept fixed. Otherwise, repeatedly

updating the well-trained networks of the lower GANs when training the upper GANs will cause an overfitting problem, and thus reduce the prediction accuracy. The training process of the HGAN-based TSA is given in Algorithm 2. For each GAN model, deploying the stochastic gradient descent algorithm, the generator and the discriminator are updated until the convergence of the cross-entropy loss, or until the training episode reaches its maximal ep_{max} , whichever occurs sooner.

Algorithm 2 The training algorithm for the developed HGAN-based TSA

Normalize the data such that the maximum normalized voltage is one **for** $i = 1, \dots, N$

do

Initialize the neural network parameters of the generator and discriminator, i.e., θ_g and θ_d in GAN i , respectively **for** $ep = 1, \dots, ep_{max}$ **do**

Obtain a minibatch of the measured post-fault PMU voltage data $x(t)$ and $x(t+i)$

Using the well-trained lower GANs k , $k \in \{1, \dots, i-1\}$ to obtain the sequence data $\{x(t), \hat{x}(t+1), \dots, \hat{x}(t+i-1)\}$ Feed the obtained sequence data to the generator G , generate the data label y , and predict data $\hat{x}(t+i)$ for the next sampling time Input the predicted data $\hat{x}(t+i)$ and real data $x(t+i)$ to the discriminator D , and obtain the estimated probability $D(\hat{x}(t+i)|\theta_d)$ and $D(x(t+i)|\theta_d)$ Update the generator and discriminator with gradient descent algorithm;

end

Save the network parameters θ_g and θ_d for GAN i

end

The trained HGAN-based TSA can be applied online for stability assessment. Once finishing training, the well-trained HGAN-based TSA can be deployed for online TSA. Upon measuring the first post-fault PMU voltage data, $v(0)$, the HGAN-based model with N sub-GANs generates the subsequent voltage data $\{v(0), \hat{v}(1), \dots, \hat{v}(N)\}$. The TSA results from these N sub-GANs will be combined by using an average-based ensemble technique for the final TSA classification. Here, it is assumed that the final TSA result

is stable if more than half of sub-GANs vote for stable; otherwise, the system is unstable. The ensemble method reduces the variance of the prediction errors in all sub-GANs and results in a more accurate TSA [135].

4.4 Simulation Results

To evaluate the performance of the HGAN-based TSA under various system operating conditions, the IEEE 118-bus system with 118 buses, 170 lines, and 9 transformers is studied. To demonstrate the effectiveness of the HGAN-based TSA method, five different baseline methods, namely, decision trees, SVM, LSTM, GRU, and stacked GRU are used for comparison purposes. Furthermore, as the TSA accuracy of the HGAN-based method relies on PMU measurements, sensitivity studies are performed to investigate the impact of noise, number, and location of PMUs on TSA accuracy. The simulations are conducted on an i7 computer with a 3.2GHz CPU and 64GB RAM.

4.4.1 PMU Data Generation

The post-fault PMU voltage time series for testing and training are generated with simulations. Various operating conditions (ranges from 90% to 110% of the nominal loading condition) are simulated under different system typologies, i.e., normal and one line out of service (due to maintenance). The contingencies considered in this section are three-phase faults on every bus and transmission line (located at 30%, 50%, and 80% of the lines). To consist with the other data-driven TSA studies, the faults are cleared after 5 cycles, and the system topology is kept unchanged after clearing the fault [121]. To minimize the adverse impacts of an imbalanced dataset on the accuracy of the HGAN-based TSA, 5,000 stable and 5,000 unstable transients are randomly picked from the generated transients. Furthermore, 80% of the data are used for training, and the remaining 20% are utilized for testing. For creating stability labels, the transient stability of each event is determined using the in-

Table 4.1: PMU placement in the IEEE 118-bus system

Test System	PMU placement
IEEE 118-bus system	5, 12, 15, 17, 32, 37, 49, 56, 59, 67, 69, 70, 71, 77, 80, 85, 92, 96, 100, 105

dex outlined in (4.2). To generate the PMU time series data, the sampling frequency of the simulation is set to 120 frames per second. Additionally, to mimic real power grids where not every bus is equipped with PMUs, 20 PMUs are deployed in the IEEE 118-bus system to ensure full observability [136]. The locations of the PMUs are given in Table 4.1, and are determined by spanning tree method. The training and testing data can be found at the IEEE DataPort (<https://dx.doi.org/10.21227/6f5v-q924>).

4.4.2 Performance Analysis of the HGAN-based TSA

—**Performance Analysis:** Upon obtaining the training and testing datasets that contain transient events, each level GAN of the developed TSA model is trained, following the training steps outlined in Algorithm 2. The training parameters are provided in Table 4.2. It is found that the HGAN-based model with three GANs is sufficient for an accurate stability assessment. Although more GANs generate longer duration of voltage time series data, and thus better represent a transient pattern, a small error in lower level GANs could propagate through the stacked GANs, and eventually lead to a large cumulative error. The learning rates of the generator and discriminator are set to 0.001 and 0.0001, respectively. A larger learning rate can increase the learning speed, while the results may converge to a suboptimal solution. Additionally, to address the overfitting problem during the training of a GAN, the neural network parameters of the lower level GANs are not updated. Overall, the training time for three sub-GANs is approximately 3.5 hours with CPU. As the training of the HGAN is only conducted once and offline, this training time is reasonable. Furthermore, the training time can be significantly reduced with GPU.

Table 4.2: Parameters of the HGAN-based TSA model

Parameters	Value
Number of GANs	3
Number of episode	20000
Learning rate of generator / discriminator	1e-3 / 1e-4
Number of GRU layers / hidden layers	2 / 30
Batch size	128

To demonstrate the learning ability of the HGAN-based model, the cross-entropy loss and the squared error of each sub-GAN in (4.9) are depicted in Fig. 4.7. To check for overfitting, the testing data is used to test the model in each training step. Note that the testing data are not used to train the HGAN-based model. As seen from Fig. 4.7, for all three GANs, the cross-entropy loss and the squared error dramatically decrease at the beginning of the training phase and converge after that. For instance, the cross-entropy loss for training data in *GAN* 1 drops from 0.7 to 0.33, and the squared error of the training data decreases from 0.1965 to 0.0002. It is also found that with the predicted data from lower GANs, the cross-entropy loss of *GAN* 2 and 3 drops faster than the loss in *GAN* 1. The faster drop indicates that combining the predicted and the measured data leads to more distinguishable transient features for the GAN model, and thus the learning speed is increased. The cross-entropy loss and the squared error reduction indicate that the developed model is effectively trained to improve the classification and prediction accuracy. Comparisons of the training and testing loss and error demonstrate that no overfitting occurs. This is concluded by the fact that the cross-entropy loss of the testing data does not rise as the loss of the training data decreases. In other words, both these values maintain a consistent difference.

Upon training the developed HGAN model, the testing data, i.e., 1000 stable and 1000 unstable events, are used to evaluate the classification accuracy of each GAN. The classification results are presented in Table 4.3. With the single measured PMU voltage data,

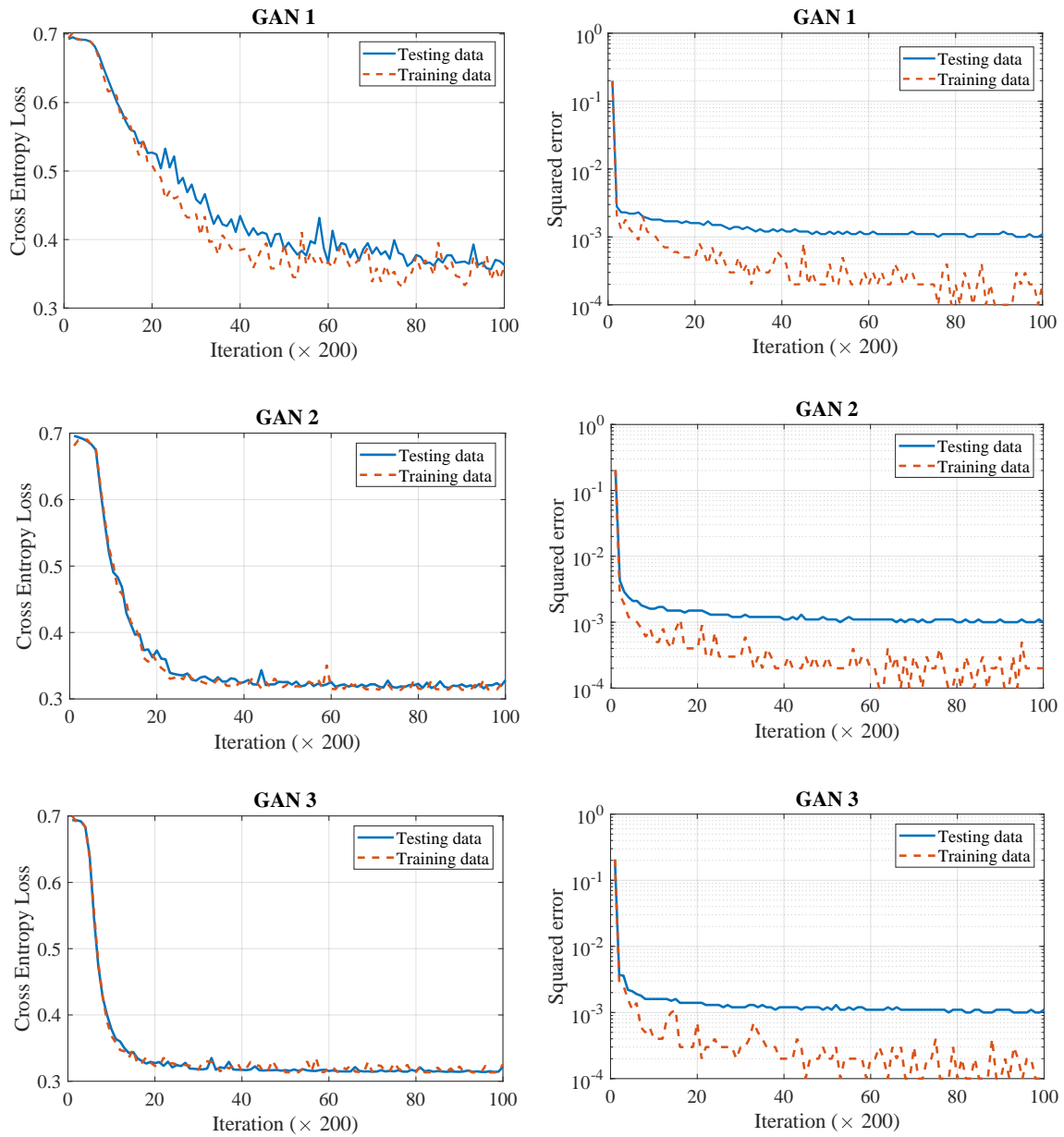


Figure 4.7: The cross-entropy loss and the squared error for each of the three GANs in the HGAN-based model.

Table 4.3: TSA classification accuracy (F1 score) of each GAN in the HGAN-based TSA

	GAN 1	GAN 2	GAN 3
Accuracy	95.88 %	99.95 %	99.95 %

Predicted Actual	GAN 1		GAN 2		GAN 3	
	Unstable	Stable	Unstable	Stable	Unstable	Stable
Unstable	100%	0%	100%	0%	100%	0%
Stable	8.6%	91.4%	0.1%	99.9%	0.1%	99.9%

Figure 4.8: The confusion matrix of each GAN in the HGAN-based TSA model

the classification accuracy of the lowest level *GAN 1* is 95.35%. However, combining the real measured data and the predicted data from the lower level GANs yields a classification accuracy of 99.95% for both *GAN 2* and *GAN 3*. Since a single sampled data carries limited transient information, the classification accuracy of *GAN 1* is only 95.3%. With more predicted data that hold the spatial and temporal features of a transient event, the hidden oscillation patterns of the event are more apparent to the classifiers and result in higher classification accuracy. The confusion matrix of each GAN is depicted in Fig. 4.8. For *GAN 1*, it can be observed that all unstable transient data are correctly identified, while 8.6% of the stable events are incorrectly classified as unstable. Benefiting from the predicted PMU voltage data from *GAN 1*, in *GAN 2*, the percentage of misclassified stable events drops to 0.1%. Combining the classification results from these three GANs with an average-based ensemble strategy yields an accuracy of 99.95% for TSA. Hence, by stacking three sub-GANs, the HGAN model achieves high classification accuracy with only one sampled PMU data. The computation time for the developed TSA with three GANs is 0.00298s (0.359 cycles). The total response time, including the waiting time for one cycle of the PMU data, is 1.359 cycles, which is reasonable for near real-time analysis.

—**Comparison with Baseline Methods:** To demonstrate the effectiveness of the developed data-driven TSA approach, it is compared with conventional data-driven methods

that deploy decision trees, SVM, LSTM, and GRU. The inputs to the baseline methods are the measured post-fault PMU voltage data. For the highest classification accuracy, the LSTM hyperparameters, namely, the learning rate, training iteration, number of layers, hidden layers, and batch size are set to 0.0005, 40000, 2, and 40, and 128 respectively. Similarly, for GRU, optimal learning rate, training iteration, number of layers, and hidden layers are found to be 0.0005, 40000, 2, and 50, respectively. The comparisons are presented in Table 4.4 and Fig. 4.9. Compared with SVM, LSTM, and GRU, the decision trees, stacked-GRU, and the HGAN-based TSA achieve higher classification accuracy, using only one sample of PMU measurements. To reach this TSA classification accuracy, the stacked GRU and HGAN-based methods require shorter response times, i.e., 1.124 and 1.359 cycles, respectively. Although the SVM, GRU, and LSTM-based TSA methods achieve the similar classification accuracy at around 2 cycles, the reduction of 0.65 cycles achieved by the developed HGAN-based method is critical for online TSA. The transient instability issues propagate rapidly in the grids. Hence, a faster TSA allows more time for mitigative actions to prevent cascading outages. The response time is the combination of the waiting time for sufficient PMU samples and the computation time of the TSA method.

To demonstrate the benefits of the discriminator in the HGAN model, a stacked GRU model, where only the generator is retained, is utilized to assess the transient stability. The settings of the stacked GRU are the same as the HGAN model. The overall classification accuracy, response time, and the classification accuracy of each sub-GRU are given in Table 4.4, Fig. 4.9 and Table 4.5, respectively. Due to the GRU structure, the stacked GRU also enables learning the temporal correlations of the sequence data, and achieves a high TSA classification accuracy, i.e., 99.1% with 1.12 cycles. However, in comparison to the TSA classification accuracy of each sub-GAN in Table 4.3, it can be found that *GAN*₂ and 3 achieve higher classification accuracy than GRU 2 and 3 in Table 4.5. The discriminator helps the generator better capture the dynamics of the transients, hence resulting in higher classification accuracy. In summary, the HGAN-based model outperforms the other

Table 4.4: TSA classification accuracy of the baseline methods and the HGAN-based TSA with only one sample of PMU measurements (20 PMUs are used)

Method	TSA Classification Accuracy
Decision tree	99.5%
SVM	95.8%
LSTM	94.75%
GRU	94.8%
Stacked GRU	99.1%
HGAN	99.95%

Table 4.5: TSA classification accuracy (F1 score) of each GRU in the stacked GRU model

	GRU 1	GRU 2	GRU 3
Accuracy	95.1 %	98.2 %	98.7 %

baseline methods from both the accuracy and applicability perspectives. It is noted that the study in [114] developed a stacked-GRU model to classify transient events. However, this stacked model is a multilayer GRU, where the higher level GRU uses the output of the lower level GRU at the same time step. This stacked GRU model is the same as the studied baseline GRU model in Table 4.4, where two layers of GRU is used. Moreover, this model can only be used for classification, and requires at least several cycles of measurements to achieve high TSA classification accuracy.

4.4.3 Impact of PMU Measurement Variations

The PMU measurements used in this section are obtained from simulations and are thus noise free. However, in practice, PMU measurements are contaminated by noise. Hence, the sensitivities of the HGAN-based TSA method to noisy PMU measurements are analyzed. Following the study in [137], and [138], white Gaussian noise is added to the simulated PMU data. Four different signal-to-noise ratios (SNR), namely, 50dB, 60dB, 70dB, and 80dB, are considered. The classification accuracies of the HGAN-based method under different noise conditions are presented in Fig. 4.10. The classification results in Fig. 4.10 show that the noise mainly affects the performance of *GAN 1*. Larger noise will lead to

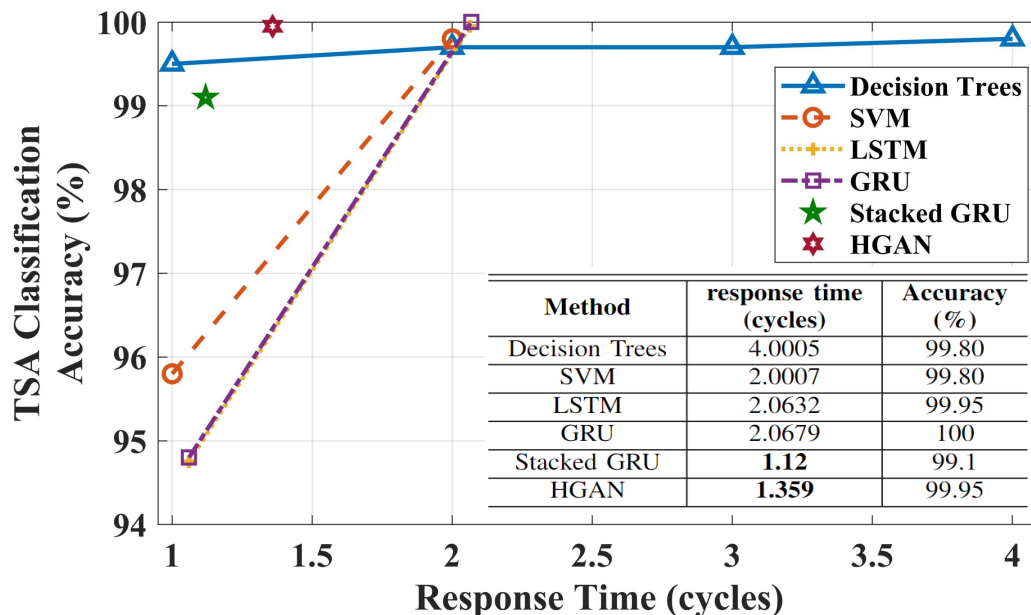


Figure 4.9: Comparison of response time between the baseline methods and HGAN-based TSA method. In total, 20 PMUs are used in the IEEE 118-bus system. To achieve the same accuracy as the HGAN-based method, other methods would use more samples of PMU measurements and thus require longer response time.

a lower classification accuracy in *GAN 1*. However, it is observed that the classification accuracy of *GAN 2* and *GAN 3* are above 99%, and are not impacted by the addition of noise. Another interesting observation is that among the five different noise conditions, *GAN 1* achieves the highest classification accuracy when a noise of 80dB SNR is added to the measurements. This accuracy is even higher than using clean PMU data. This observation can be explained by the known fact that adding a small noise to the input data leads to a better performance of the GAN models, as they learn more general features of the transient data for a better classification performance [139].

In practice, PMUs are not deployed on every bus. Hence, the performance of the HGAN-based TSA varies by location. To evaluate the performance of the developed approach under varying PMU placement scenarios, five different PMU placement scenarios are investigated. The PMU locations for these five scenarios are listed in Table 4.6. The classification accuracy for each GAN under these scenarios is depicted in Table 4.7. As seen, the

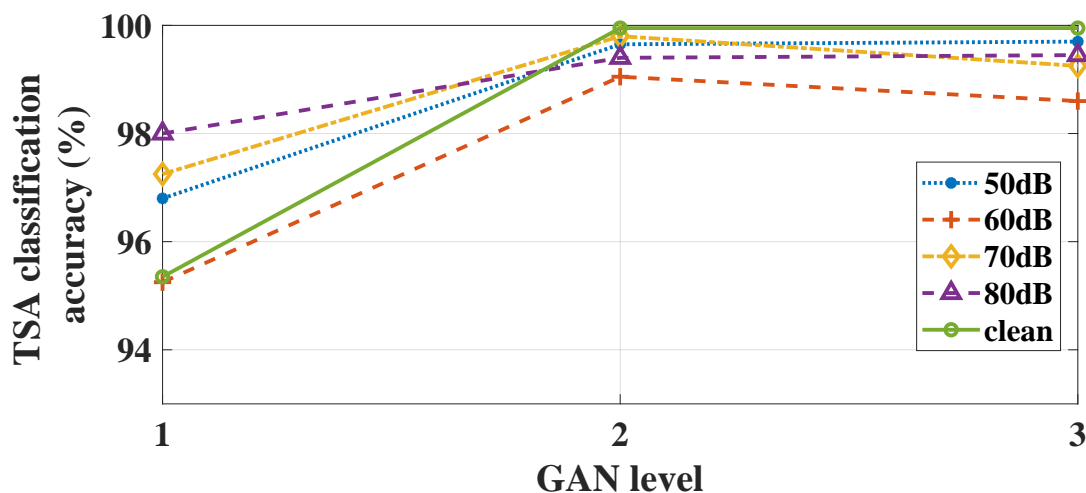


Figure 4.10: The TSA classification accuracy of each sub-GAN under different noise conditions. The testing data are used for evaluation.

Table 4.6: PMU location scenarios for investigating the impact of PMU placement on the developed TSA approach

Scenario	Location
1	5, 12, 15, 17, 37, 49, 59, 69, 71, 77, 80, 85, 92, 96, 100, 105
2	5, 12, 15, 17, 37, 56, 67, 69, 70, 71, 77, 80, 92, 96, 100, 105
3	5, 12, 15, 17, 37, 49, 67, 69, 70, 71, 77, 80, 92, 96, 100, 105
4	5, 12, 15, 17, 32, 37, 49, 56, 59, 71, 77, 80, 85, 92, 96, 100
5	5, 12, 15, 17, 32, 37, 49, 56, 59, 69, 71, 77, 80, 92, 96, 100

location of PMUs significantly impacts the classification performance of *GAN1*, where the accuracy ranges from 94.55% to 97.9%. Benefiting from the predicted data, the impact of placement is decreased in *GAN2* and *GAN3*. Moreover, the classification accuracy slightly drops in *GAN3*, due to the accumulation of small prediction errors in lower GANs.

The impact of the number of PMUs is demonstrated in Table 4.8. Increasing the number of PMUs improves the TSA accuracy for *GAN1*. At 12 and above PMUs, the classification accuracy of *GAN2* and 3 are around 99%. However, with less than 10 PMUs the classification accuracy of *GAN2* and *GAN3* drops, indicating that the predicted data from the lower GANs are not close to real data. The prediction error propagates to higher GANs and increases cumulative error, which degrades the classification accuracy. For example, with 8 PMUs, the classification accuracy is 91.15% in *GAN2* and drops to 77.95% in *GAN3*.

Table 4.7: The TSA classification accuracy of each GAN with different PMU locations. The testing data are used for evaluation. A total number of 16 PMUs are used

PMU Location Scenario	TSA Classification Accuracy (%)		
	GAN 1	GAN 2	GAN 3
1	94.55	98.55	97.90
2	97.15	99.40	98.40
3	96.65	99.65	99.15
4	97.90	98.55	97.55
5	95.40	98.25	97.20

Table 4.8: TSA classification accuracy of each sub-GAN with different number of PMUs

Number of PMUs	GAN 1	GAN 2	GAN 3
6	86.90%	84.65%	88.20%
8	91.20%	91.15%	77.95%
10	94.20%	94.55%	95.20%
12	95.75%	99.60%	99.65%
14	96.05%	99.93%	99.92%
16	96.65%	99.65%	99.15%
18	96.25%	99.40%	99.91%
20	95.35%	99.95%	99.95%

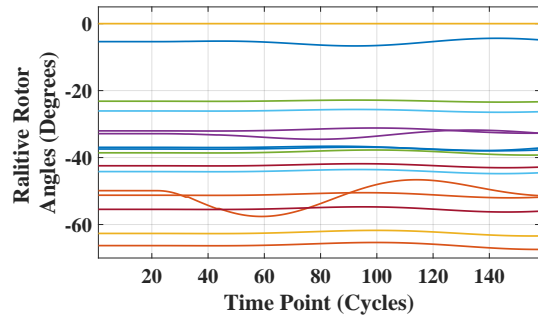
While more PMU data improve the classification accuracy, when the number of PMUs is larger than a threshold, e.g., 12 in the IEEE 118-bus system, the classification accuracy remains fixed in the higher level GANs.

4.5 Discussions

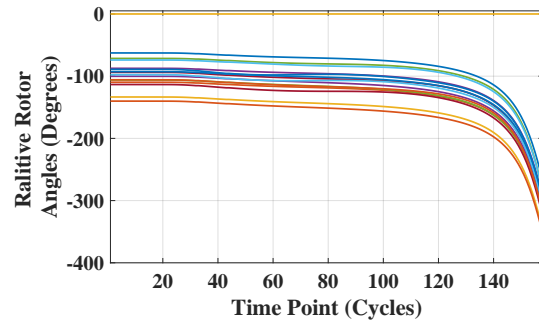
Examples of the predicted post-fault voltage time series for both stable and unstable transients are shown in Fig. 4.11. The relative rotor angles of the stable and unstable transient events are shown in Fig. 4.11a and Fig. 4.11b, respectively. The HGAN-based TSA has five GANs, and thus can predict the voltage of the subsequent five time steps $\{\hat{v}(2), \dots, \hat{v}(6)\}$, using the measured post-fault voltage data $v(1)$. As observed in Fig. 4.11g and Fig. 4.11h, the average prediction errors for both stable and unstable transients are within 0.03 pu. By only using the measured PMU voltage data from the first

time instant, i.e., $v(1)$, the HGAN-based model predicts the oscillation features of these two events, and for the most part, generates similar time series data as the real measurements. With the prediction capability, the developed approach can also be deployed in other applications (e.g., predicting stability margins) that benefit from the ability to predict how transients progress. The prediction accuracy of the sequence data relies on the performance of each sub-GAN. The error in lower level GANs propagates to higher level GANs and results in a large accumulated error. To improve the prediction, we suggest: (1) freezing the lower-level GANs when training the higher-level GANs to avoid overfitting; (2) adding additional encoder and decoder to generator to better predict the sequence data.

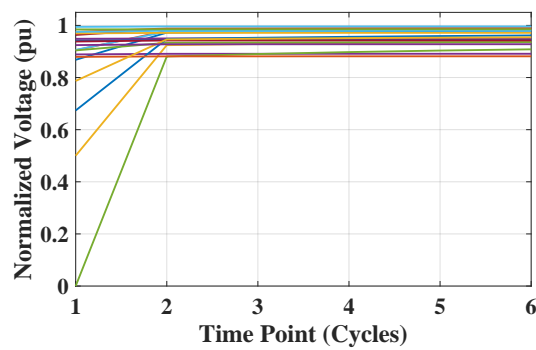
The performance of the developed TSA approach naturally relies on the training data. As the transient data are generated from simulations, the training dataset is ensured to be balanced, i.e, the number of stable and unstable transients is close. However, the real-world transient training datasets are imbalanced, i.e., the ratio between the unstable and stable events is less than 5%, since the unstable transient events are low-frequency events. An imbalanced training dataset may adversely affect the classification accuracy. To address the problem of imbalanced data, two strategies, i.e., data resampling and redesigning the classifier algorithm, are suggested to improve the HGAN-based TSA method. The data resampling methods, such as oversampling and undersampling methods, increase the ratio between the two types of events [140]. In addition to resampling, the classifier algorithm can be adjusted, such that it is more sensitive to unstable events. For instance, in [141], the misclassification cost is modified such that a higher cost is assigned to the wrong prediction of the unstable events.



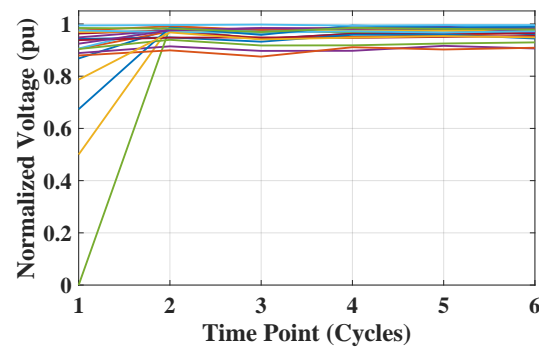
(a) Relative rotor angles of a real stable transient



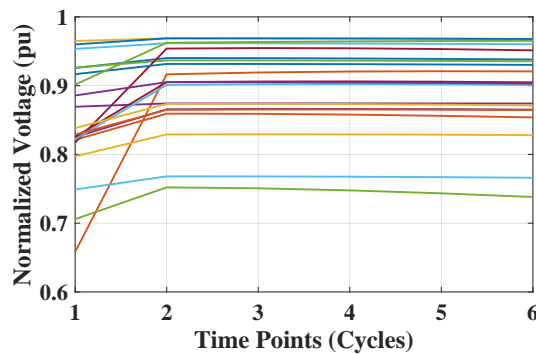
(b) Relative rotor angles of a real unstable transient



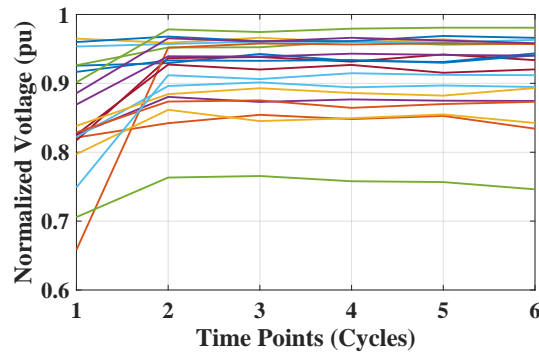
(c) Real stable transient



(d) Predicted stable transient

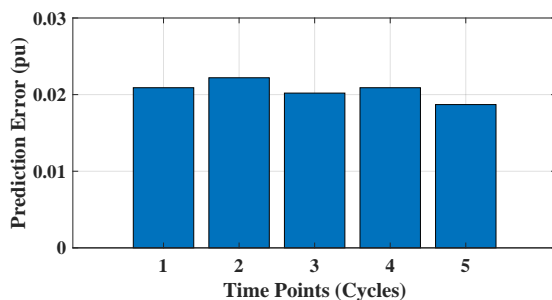


(e) Real unstable transient

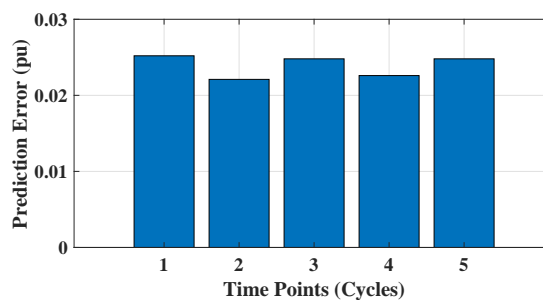


(f) Predicted unstable transient

Figure 4.11: Comparison between the real and predicted post-fault stable and unstable transients. The developed TSA approach has 5 GANs and the number of PMUs is 20. The PMU voltage measurements are normalized using a Min-Max scaling method.



(g) Average prediction error of stable transient



(h) Average prediction error of unstable transient

Figure 4.11: Comparison between the real and predicted post-fault stable and unstable transients. The developed TSA approach has 5 GANs and the number of PMUs is 20. The PMU voltage measurements are normalized using a Min-Max scaling method.

4.6 Conclusions

This section developed a novel GAN-based TSA approach to promptly and accurately assess the post-fault transient status of a power system using PMU-provided voltage measurements. Unlike the conventional GAN model, the generator is redesigned such that the modified GAN model predicts the sequence data from real-time measurements. Benefiting from the specific hierarchical structure of multiple GANs, the developed HGAN-based model maintains the spatial and temporal features of the multivariate PMU time series data. Therefore, with only one sample of PMU measurement, the developed TSA approach improves the stability assessment accuracy. The ability to utilize only one sample of data to generate accurate predictions of future system transients is a significant achievement compared to the conventional methods. Case studies conducted on the IEEE 118-bus system demonstrate that the TSA accuracy can reach 99.95%, specifically using the predicted data. Compared with the other four machine-learning-based methods, i.e., decision trees, SVM, GRU, and LSTM, and the stacked GRU model, the HGAN-based TSA achieves a higher classification accuracy and is faster. To demonstrate the robustness of the HGAN-based TSA, the impacts of measurement noise, location, and the number of available measurements are investigated. It is observed that the prediction capability

of the HGAN-based TSA increases the robustness to these variations while improving the classification accuracy. Following proper offline training, the developed TSA can be applied in near real-time to assess the post-fault stability of a power system. Additionally, as the HGAN model enables learning the spatial and temporal features of the system transients, the predicted sequence data can be used for various applications that benefit from the ability to predict the transient response of a system.

Chapter 5

Impact Mitigation of Power System

Failures

5.1 Introduction

The aforementioned chapters have analyzed event detection and impact prediction. After the event is detected, it is also critical to promptly mitigate failures, and prevent the propagation of cascades. This chapter is concerned with the optimal mitigation of the power system failures, in particular, thermal overloads. With the large-scale adoption of phasor measurement units, new and more effective mitigation opportunities have emerged. However, the possibility of false data attacks on the measured data, or the relay status information, threatens the promise of utilizing measurements for timely mitigation of thermal overloads. False data can lead to the wrong estimation of the system states and the power flow model, which may eventually lead to wrong mitigations. Authors in [2, 142–145] have shown that malicious false data can be injected into PMU measurements through various communication protocols such as IEC 61850 and IEC 60870-5-101, and remain undetected.

To address these challenges, a data-driven model-predictive control (MPC) method is introduced for mitigation of the thermal overloads, which is resilient to false data injection attacks. This is achieved by constructing a data-driven power flow model from the trustworthy system measurements, that is independent from system topology and model parameters. Hence, the power flow model becomes immune to wrong system model infor-

mation. Additionally, to eliminate the adverse impacts of false data on the measurements, the actual system states are recovered from the historical trustworthy data if an attack is detected.

5.2 Background

Various overload mitigation methods have been developed, which can be grouped into two main categories of transmission line switching [146–148], and power re-dispatch methods [149–152]. Line switching methods are cost-effective, and mitigate overloads via switching transmission lines and bus-bars. However, they are computationally burdensome. The power re-dispatch methods, solve a carefully formulated optimal power flow (OPF) to identify feasible mitigations with a reasonable computational complexity. However, the open loop feature and the one-step ahead control actions limit the applicability of these OPF based re-dispatch methods.

To address the limitations of the aforementioned methods, MPC has been utilized as an alternative approach to identify optimal control actions. MPC has been applied in power systems for various purposes, such as voltage control [153, 154], frequency control [155, 156], and overload mitigation [157–162]. Unlike the conventional one-step ahead controls, MPC-based controllers determine the optimal actions based on projections of their impact on the proceeding finite time-steps [163]. The optimization process is repeated for the subsequent time steps to identify the corresponding corrective actions based on the actual system conditions. With a closed-loop feedback feature, MPC can handle unexpected system operating conditions and measurement noise [157].

The MPC-based overload mitigation strategies developed in [157, 158] are improved in [159] by considering the temperature limits of the lines, which lead to dynamic line ratings. Dynamic line ratings more accurately reflect thermal constraints than the static line ratings.

The MPC-based overload mitigation methods are further expanded in [160–162], utilizing other types of controllable resources, such as energy storage and distributed energy resources.

5.3 Power Flow Model

5.3.1 Model-based Linear DC Power Flow Model

The model-based DC power flow model is linear and computationally efficient in identifying optimal mitigative actions with the MPC method [161]. For a power system with N_{bus} buses and N_{line} lines, three assumptions are made in a DC power flow model [164]: 1) line resistances are neglected; 2) all bus voltage magnitudes are one per unit; 3) the phase angles of the two neighboring buses are close. Under these assumptions, the relationship between the active power flow of lines and the active power injection at buses can be described by a power transfer distribution factor (PTDF) matrix $\Psi \in \mathbb{R}^{N_{\text{line}} \times N_{\text{bus}}}$ as,

$$\Psi = \left[\mathbb{O}, B\tilde{A}(\tilde{A}^T B\tilde{A})^{-1} \right], \quad (5.1)$$

where $\mathbb{O} \in \mathbb{R}^{N_{\text{line}}}$ is an all zeros column vector. The matrix \mathbf{B} is a diagonal susceptance matrix, where $B_{ii} = b_i$ and b_i is the susceptance of line i . The matrix $\tilde{A} \in \mathbb{R}^{N_{\text{line}} \times (N_{\text{bus}} - 1)}$ is a reduced-sized incidence matrix of a power system, which is obtained from the incidence matrix $\mathbf{A} \in \mathbb{R}^{N_{\text{line}} \times N_{\text{bus}}}$ by removing the column that corresponds to the slack bus. Given the system topology information, the edge-to-node incidence matrix \mathbf{A} is obtained as,

$$A_{ij} = \begin{cases} 1 & \text{If bus } j \text{ is the } \textit{from} \text{ bus in line } i \\ -1 & \text{If bus } j \text{ is the } \textit{to} \text{ bus in line } i \\ 0 & \text{otherwise} \end{cases} . \quad (5.2)$$

The relationship between the active power flow of line i and active power injections at all buses is [165],

$$Pf_i = \Psi_i \cdot Pinj, \quad \forall i \in \{\text{Lines}\}, \quad (5.3)$$

Given equation (5.3), a linear DC power flow model can be formulated as a linear time invariant (LTI) system [166]. To match the sampling time of the system meters, the LTI system is first discretized as,

$$Pf_i[k+1] = Pf_i[k] + \Psi_i \cdot \Delta Pinj[k] \quad (5.4.1)$$

$$Pf_i^{\text{mea}}[k] = Pf_i[k] + \varepsilon_i[k], \quad \forall i \in \{\text{Lines}\}, \quad (5.4.2)$$

where the power injection change at time k , $\Delta Pinj[k] \in \mathbb{R}^{N_{\text{bus}}}$, is defined as,

$$\Delta Pinj[k] = \mathbf{A}_{\text{bus}}^{\text{gen}} \Delta G[k] - \mathbf{A}_{\text{bus}}^{\text{load}} \Delta D[k], \quad (5.5)$$

The entry in the i^{th} row and j^{th} column of $\mathbf{A}_{\text{bus}}^{\text{gen}}$ ($\mathbf{A}_{\text{bus}}^{\text{load}}$) is one, if the j^{th} generator (load) is located at bus i , and zero otherwise. The input to the LTI system in (5.6) is the active power injection change $\Delta Pinj \in \mathbb{R}^{N_{\text{bus}}}$, and the output is the power flow of line i , Pf_i^{mea} , $\forall i \in \{\text{Lines}\}$. For brevity, we use Pf_i to denote the output of the LTI system for the rest of this chapter. The LTI system in (5.6) is controllable as the controllability matrix $\Omega = \Psi_i$, has a full row rank [167]. Furthermore, incorporating (5.5) into this LTI system can lead to:

$$\begin{aligned} \begin{bmatrix} Pf[k+1] \\ G[k+1] \\ D[k+1] \end{bmatrix} &= \begin{bmatrix} \Psi \cdot \mathbf{A}_{\text{bus}}^{\text{gen}} & -\Psi \cdot \mathbf{A}_{\text{bus}}^{\text{load}} \\ \mathbb{I}_g & \mathbb{O}_d \\ \mathbb{O}_g & \mathbb{I}_d \end{bmatrix} \begin{bmatrix} \Delta G[k] \\ \Delta D[k] + \Delta D^{\text{var}}[k] \end{bmatrix} \\ &+ \begin{bmatrix} Pf[k] \\ G[k] \\ D[k] \end{bmatrix} + \begin{bmatrix} e_{pf}^1[k] \\ e_g^1[k] \\ e_d^1[k] \end{bmatrix} \end{aligned} \quad (5.6.1)$$

$$\begin{bmatrix} Pf^{mea}[k] \\ G^{mea}[k] \\ D^{mea}[k] \end{bmatrix} = \begin{bmatrix} Pf[k] \\ G[k] \\ D[k] \end{bmatrix} + \begin{bmatrix} e_{pf}^2[k] \\ e_g^2[k] \\ e_d^2[k] \end{bmatrix} \quad (5.6.2)$$

where $\mathbb{I}_g \in \mathbb{R}^{N_{gen} \times N_{gen}}$ and $\mathbb{I}_d \in \mathbb{R}^{N_{load} \times N_{load}}$ are identity matrices. $\mathbb{O}_g \in \mathbb{R}^{N_{gen} \times N_{gen}}$ and $\mathbb{O}_d \in \mathbb{R}^{N_{load} \times N_{load}}$ are zero matrices. $Pf^{mea}[k]$, $G^{mea}[k]$ and $D^{mea}[k]$ are the measured power flow, generation, and load at time k , respectively. The vectors e_{pf}^1 , e_g^1 , e_d^1 and e_{pf}^2 , e_g^2 , e_d^2 are the noise vectors, due to load prediction and metering errors, for the LTI system.

5.3.2 Parameter-based Power Flow Model

The previous section introduces the model-based power flow. It can be observed in (5.6) that the model-based linear power flow model requires accurate system model information. An error in the topology processor can lead to a wrong incidence matrix \mathbf{A} or a wrong susceptance matrix \mathbf{B} . As a result, an incorrect PTDF matrix Ψ is achieved, which in turn leads to a wrong estimation of the impacts of power injection changes on line power flows. To address the problem in conventional model-based power flow, a data-driven power flow model is developed, which is illustrated in Fig. 5.1. For the discrete LTI system described in (5.6), the input column vector at time k , which is the change of the power injection, is defined as $\Delta Pinj(k) = [\Delta Pinj_1(k), \dots, \Delta Pinj_{N_{bus}}(k)]^T$. Given a sequence of the input vectors from time zero to $N - 1$, i.e., $\{\Delta Pinj(u)\}_{u=0}^{N-1}$, these sequence input vectors, i.e., input trajectory, can be represented as a stacked column vector $\Delta Pinj_{[0, N-1]} = [\Delta Pinj^T(0), \dots, \Delta Pinj^T(N - 1)]^T$.

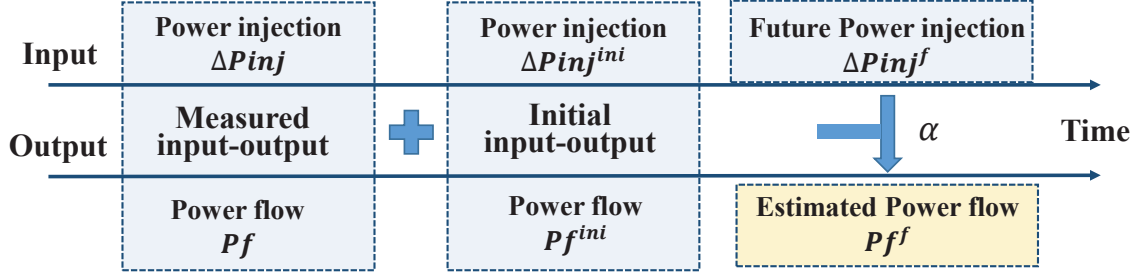


Figure 5.1: The developed data-driven power flow model

Assume the length of the prediction horizon and the input data are L and N ($N \geq L$), respectively. Then, the Hankel matrix of this trajectory is,

$$\mathbf{H}_L(\Delta Pinj_{[0, N-1]}) = \begin{bmatrix} \Delta Pinj(0) & \Delta Pinj(1) & \cdots & \Delta Pinj(N-L) \\ \Delta Pinj(1) & \Delta Pinj(2) & \cdots & \Delta Pinj(N-L+1) \\ \vdots & \vdots & \ddots & \vdots \\ \Delta Pinj(L-1) & \Delta Pinj(L) & \cdots & \Delta Pinj(N-1) \end{bmatrix}. \quad (5.7)$$

If the Hankel matrix has a full row rank, the stored data in \mathbf{H}_L can be used to learn the power flow model [168]. In other words, in order for the rows of the Hankel matrix to be independent (full rank), the sequence $\{\Delta Pinj_u\}_{u=0}^{N-1}$ should be sufficiently long and rich [168], which implies that $N \geq 2L - 1$.

Lemma 1 *Given the observed input-output sequences of the LTI system described in (5.6), i.e. $\{\Delta Pinj(u), Pf_i(u)\}_{u=0}^{N-1}$, the sequence $\{\Delta \overline{Pinj}(v), \overline{Pf}_i(v)\}_{v=0}^{L-1}$ is also the trajectory of the same system if and only if there exists an $\alpha \in \mathbb{R}^{N-L+1}$ such that,*

$$\begin{bmatrix} \mathbf{H}_L(\Delta Pinj_{[0, N-1]}) \\ \mathbf{H}_L(Pf_{i, [0, N-1]}) \end{bmatrix} \alpha = \begin{bmatrix} \Delta \overline{Pinj}_{[0, L-1]} \\ \overline{Pf}_{i, [0, L-1]} \end{bmatrix}. \quad (5.8)$$

Lemma 1, which is proven for an LTI system in [168], is extended to the problem considered here. It indicates that, for an unknown LTI system, any L -long system trajectory can be represented by the observed data, which is the foundation of the developed data-driven power flow model.

It can be found in Lemma 1 that, in comparison with the parametric data-driven system identification methods that require additional steps to calculate system state, input and output matrices, this non-parametric method deploys the Hankel matrix to represent the system model directly using the system data. Lemma 1 requires the length of the data sequence $\{\Delta Pinj(u), Pf_i(u)\}_{u=0}^{N-1}$ to be large enough to meet the constraint: $N \geq N_{\text{bus}}(L + 1) + L$.

From (5.8) the output trajectory $\overline{Pf}_{i, [0, L-1]}$ can be computed by first solving $\mathbf{H}_L(\Delta Pinj_{[0, N-1]})\alpha = \Delta \overline{Pinj}_{[0, L-1]}$ for α , and then solving $\overline{Pf}_{i, [0, L-1]} = \mathbf{H}_L(Pf_{i, [0, N-1]})\alpha$. To further estimate the impact of the power injection changes on a specific power flow, the initial condition of the power flow $\{\overline{Pf}_i(v)\}_{v=0}^{L-1}$, which is the first N_{ini} data points $\{\overline{Pf}_i(v)\}_{v=0}^{N_{\text{ini}}-1}$, has to be specified [169]. The Hankel matrices of the observed input-output trajectories $\mathbf{H}_L(\Delta Pinj)$ and $\mathbf{H}_L(Pf_i)$, after defining the length of the initial trajectory N_{ini} , can then be partitioned as $\mathbf{H}_{N_{\text{ini}}+N_f}(\Delta Pinj) = [\mathbf{H}_{N_{\text{ini}}}^{\text{ini}}(\Delta Pinj), \mathbf{H}_{N_f}^f(\Delta Pinj)]^T$ and $\mathbf{H}_{N_{\text{ini}}+N_f}(Pf) = [\mathbf{H}_{N_{\text{ini}}}^{\text{ini}}(Pf_i), \mathbf{H}_{N_f}^f(Pf_i)]^T$.

Lemma 2 *Given: (a) the observed input-output trajectories of a power system, i.e., $\{\Delta Pinj(u), Pf_i(u)\}_{u=0}^{N-1}$; (b) the initial condition of the system $\{\Delta \overline{Pinj}^{\text{ini}}(u), \overline{Pf}_i^{\text{ini}}(u)\}_{u=0}^{N_{\text{ini}}-1}$, and (c) the future power injection change $\{\Delta \overline{Pinj}^f(u)\}_{u=0}^{N_f-1}$, if the Hankel matrix of the input trajectory $\Delta Pinj_{[0, N-1]}$ has full row rank of $N_{\text{ini}} + N_f + 1$ and $N_{\text{ini}} \geq 1$, there exists*

an $\alpha \in \mathbb{R}^{N-N_{ini}-N_f+1}$ such that,

$$\begin{bmatrix} \mathbf{H}_{N_{ini}}^{ini}(\Delta Pinj) \\ \mathbf{H}_{N_f}^f(\Delta Pinj) \\ \mathbf{H}_{N_{ini}}^{ini}(Pf_i) \end{bmatrix} \alpha = \begin{bmatrix} \overline{\Delta Pinj}_{[0, N_{ini}-1]}^{ini} \\ \overline{\Delta Pinj}_{[0, N_f-1]}^f \\ \overline{Pf}_i^{ini}_{[0, N_{ini}-1]} \end{bmatrix}. \quad (5.9)$$

The power flow of line i , due to the power injection change $\{\overline{\Delta Pinj}^f(u)\}_{u=0}^{N_f-1}$, can then be estimated as $\overline{Pf}_i^f_{[0, N_f-1]} = \mathbf{H}_{N_f}^f(Pf_i) \cdot \alpha$.

Lemma 2 is proven in [169] for LTI systems. Here, it is extended to the data-driven power flow model so that with the initial input-output trajectory, the future trajectory is uniquely determined. Due to the presence of measurement noise, and because equation (5.9) is an under-determined equation, α can be obtained via regularized least square method,

$$\alpha = (\mathbf{H}_{ini-f}^T \mathbf{H}_{ini-f} + \mu \mathbf{I})^{-1} \mathbf{H}_{ini-f}^T D_{ini-f}. \quad (5.10)$$

where $\mathbf{H}_{ini-f} = [\mathbf{H}_{N_{ini}}^{ini}(\Delta Pinj), \mathbf{H}_{N_f}^f(\Delta Pinj), \mathbf{H}_{N_{ini}}^{ini}(Pf)]^T$ and $D_{ini-f} = [\overline{\Delta Pinj}^{ini}, \overline{\Delta Pinj}^f, \overline{Pf}_i^f]^T$ are the Hankel matrix and the data vector in (5.9), respectively. Lemma 2 and (5.10) demonstrate how the power flows can be obtained from the measurements, regardless of the system topology information or transmission line parameter errors. In this chapter, this data-driven power flow model is leveraged to determine proper overload mitigations, and amend malicious data using stored trustworthy data. Note that, it is assumed that the stored historical PMU data have been validated by an attack detection method and are thus trustworthy.

5.4 Recovery-based Model-driven MPC model

A model predictive cascade mitigation methodology is developed in this section to optimize the controls over a finite time horizon, starting from the current time instant k . Unlike

the conventional control methods, MPC method enables handling all system constraints, e.g. power flow and generation limits, and incorporates the evolution of the system state to decide on control actions [163]. Thus, based on the predicted system state at time instant k , an RMPC model is formulated to eliminate the system violations once system component failures are detected. Additionally, the RMPC model is further improved by addressing the issue of detection delays caused by the data anomaly detection methods.

5.4.1 State Recovery

When the state of system variables at time k , is detected to be incorrect due to cyber-physical attacks, the historical trustworthy data of the system will be utilized to recover the compromised state variables. This concept, referred to as *physical state recovery*, rolls the state of the system forward to the time instant k , starting from a stored state at time instant $k-N$ [170]. Instead of using the wrong states at time k , the states are predicted by using the historical stored data between time instant $k-N$ to k . Given the stored data at time $k-N$ and changes of the power injections from time $k-N$ to $k-1$, the power flow of lines can be predicted as,

$$\widehat{Pf}[k] = \widetilde{Pf}[k-N] + \sum_{i=1}^N PTDF \cdot \Delta Pinj[k-i] + \sum_{i=1}^N e_{pf}^1[k-i] \quad (5.11)$$

where $\widehat{Pf}[k]$ is the predicted power flow at time k using historical data. The trustworthy state $\widetilde{Pf}[k-N]$ is estimated from the measured power flow at time $k-N$, i.e. $Pf^{mea}[k-N]$. It can be observed that the prediction error increases with the length of the prediction horizon N . Larger prediction horizon N would lead to a less accurate predicted power flow at time k . After predicting the power flows, the incorrect states are replaced with the predicted states. Assume at time k , N_{attack} number of power flows are detected as compromised states. Thus, the system states at time k can be divided into two sets of states, i.e., attacked states $\mathbf{Pf}^{attack}[k] = \{Pf_1^{attack}[k], \dots, Pf_{N_{attack}}^{attack}[k]\}$ and trusted

states $\mathbf{Pf}^{trust}[k] = \{\widetilde{Pf}_1^{trust}[k], \dots, \widetilde{Pf}_{N_{branch}-N_{attack}}^{trust}[k]\}$. With the state recovery (5.11), the incorrect power flows are replaced with predicted results, denoted as $\mathbf{Pf}^{predict} = \{Pf_1^{predict}, \dots, Pf_{N_{attack}}^{predict}\}$. The power flow states that are used for deciding control actions are thus represented as $\mathbf{Pf}^{amend}[k] = \{\mathbf{Pf}^{trust}[k] \cup \mathbf{Pf}^{predict}[k]\}$. With same state recovery process, the amended state of load can be obtained and is represented as $\mathbf{D}^{amend}[k] = \{\mathbf{D}^{trust}[k] \cup \mathbf{D}^{predict}[k]\}$. Note that, as explained in [2], the generation measurements are assumed to be non-attackable due to the strict security measures present. It is also assumed that the historical input data, i.e. $\Delta Pinj$ between time $k-N$ to $k-1$ and the system state at time $k-N$ have already passed attack data detection process and are safely stored.

5.4.2 RMPC Model Construction

The RMPC model determines the control actions by solving the optimization problem presented in (5.12) over a finite time horizon, where the initial system states are obtained from the predicted states using (5.11) if data anomalies are detected. Based on the constructed LTI model in (5.6), RMPC predicts the evolution of power system states over time, and thus can optimize the control action sequence in a desired prediction horizon. Given the length of the prediction horizon N^{RMPC} , the RMPC formulation is defined as follows,

$$\text{Minimize}_{\Delta G[l], \Delta D[l]} \sum_{l=0}^{N^{RMPC}-1} (\Delta G^T[l] \cdot R \cdot \Delta G[l] \quad (5.12.1)$$

$$+ \Delta D^T[l] \cdot Q \cdot \Delta D[l] + \Delta S^T[l] \cdot SL \cdot \Delta S[l]$$

s.t.

$$Pf[l+1] = Pf[l] + PTDF \cdot \Delta Pinj[l] \quad (5.12.2)$$

$$G_g[l+1] = G_g[l] + \Delta G_g[l] \quad (5.12.3)$$

$$D_d[l+1] = D_d[l] + \Delta D_d[l] + \Delta D_d^{var}[l] \quad (5.12.4)$$

$$\sum_{g=1}^{N_{gen}} G_g(l+1) - \sum_{d=1}^{N_{load}} D_d(l+1) = 0 \quad (5.12.5)$$

$$|Pf[l]| \leq Pf^{max} + S[l] \quad (5.12.6)$$

$$S[l] \geq 0 \quad (5.12.7)$$

$$-G_g^{down} \leq \Delta G_g[l] \leq G_g^{up} \quad (5.12.8)$$

$$G_g^{min} \leq G_g[l] \leq G_g^{max} \quad (5.12.9)$$

$$0 \leq \Delta D_d[l] \leq D_d^{shed} \quad (5.12.10)$$

$$\sum_{t=0}^l \Delta D_d[t] \leq D_d^{shed} \quad (5.12.11)$$

$$|Pf[N^{RMPC}]| \leq Pf^{max} \quad (5.12.12)$$

where $\Delta G[l] \in \mathbb{R}^{N_{gen}}$ and $\Delta D[l] \in \mathbb{R}^{N_{load}}$ are the generation ramping and load shedding control actions at the prediction time instant l . N_{gen} and N_{load} are the number of generators and loads. The weight matrices $R \in \mathbb{R}^{N_{gen} \times N_{gen}}$ and $Q \in \mathbb{R}^{N_{load} \times N_{load}}$ are used to penalize the generation ramping and load shedding, respectively. To obtain a feasible solution from the optimization problem in (5.12), the slack variable $S[l] \in \mathbb{R}^{N_{branch}}$ is introduced to convert the hard constraint of line power flows, i.e. $|Pf[l]| \leq Pf^{max}$, into a soft constraint, that is, $|Pf[l]| \leq Pf^{max} + S[l]$, where the slack variable is non-negative and is penalized in the objective. The rationale for using the slack variables is that, when the power flow of a transmission line is above but is not far from its limit, this line can withstand the overloads for a short duration of time. The penalty weight matrix is $SL \in \mathbb{R}^{N_{branch} \times N_{branch}}$, where N_{branch} is the number of the transmission lines. Pf^{max} is the maximum capacity of the lines. The load variation at time l is represented as $\Delta D_d^{var}[l]$. The power injection change at prediction time l is the combination of generation ramping, load shedding and load variation, where the power injection at bus n can be computed by solving (5.5). G_g^{down} and G_g^{up} are the ramping down and up limits for the g^{th} generator. The maximum and minimum generation of the g^{th} generator are denoted as G_g^{max} and G_g^{min} . Since some of the loads are critical (e.g. hospitals) and cannot be shed, we assume partial load shedding. Thus, the amount of available load that can be shed at d^{th} load is represented as D_d^{shed} .

The objective of the RMPC model (5.13.1) is to minimize the total cost of the control actions and the slack variables over the finite time horizon $[k, k + N^{RMPC}]$. The constraints (5.13.2)-(5.12.4) calculate the the power flow, generation and load at the prediction time $l+1$. The supply and demand balance at each prediction time instant is guaranteed in (5.12.5). The constraint of power flow is converted to a soft constraint by introducing the slack variable S and is given in (5.13.3). The non-negative constraint of slack variable is given in (5.12.7). The limitations of generation ramping and load shedding are given in (5.12.8) and (5.12.10), and the generation limit of a generator is guaranteed in (5.12.9). The constraint (5.12.11) is to guarantee the total load lost in a load bus is within its maximum limit. The terminal constraint (5.13.4) is used to guarantee that all overloads are eliminated at the end of the prediction time.

After obtaining the control actions for a prediction horizon, only the first step of the optimized actions is applied as a control action, while the entire process is repeated to find the control actions for the next time step. In this section, the control actions are load shedding and generation ramping up/down, and the objective of the RMPC is to eliminate system limit violations, in particular, transmission line overloads. The steps of the developed RMPC scheme are as follows,

Step 1 At time k , estimate system variables $\widetilde{P}f[k]$, $\widetilde{G}[k]$ and $\widetilde{D}[k]$ from the measured data.

If the estimated variables pass the attack detection method, set the states at time k as $\mathbf{P}f^{\text{initial}}[k] = \widetilde{P}f[k]$, $\mathbf{G}^{\text{initial}}[k] = \widetilde{G}[k]$, $\mathbf{D}^{\text{initial}}[k] = \widetilde{D}[k]$ and go to step 3; otherwise, go to step 2;

Step 2 Deploying (5.11), the historical trustworthy power injection changes $\Delta Pinj[k - i]$, $i \in \{1, 2, \dots, N\}$, and the system state at time instant $k-N$, to amend wrong system states at time instant k . After recovering the system states at time k , set the system states as $\mathbf{P}f^{\text{initial}}[k] = \mathbf{P}f^{\text{amend}}[k]$, $\mathbf{D}^{\text{initial}}[k] = \mathbf{D}^{\text{amend}}[k]$ and go to step 3;

Step 3 With the initial states $\mathbf{P}f^{initial}[k]$, $\mathbf{G}^{initial}[k]$ and $\mathbf{D}^{initial}[k]$, solve the MPC optimization problem (5.12) over a defined prediction horizon $[k, k + N^{RMPC}]$ and obtain the optimized control actions, i.e., $\Delta G[l|k]$, $\Delta D[l|k]$, $l \in \{0, 1, \dots, N^{RMPC} - 1\}$, where $\Delta G[l|k]$ and $\Delta D[l|k]$ denote the generation ramping up/down and load shedding at the prediction time instant $k+l$. For simplicity, we use $\Delta G[l]$ and $\Delta D[l]$ to denote $\Delta G[l|k]$ and $\Delta D[l|k]$;

Step 4 Apply the first obtained control action, that is, $\Delta G[0]$ and $\Delta D[0]$;

Step 5 Set $k = k + 1$ and back to step 1.

5.4.3 Improving RMPC Scheme with Checkpointing Protocol

The aforementioned RMPC method assumes that the lag between the injection and the detection of an attack data is negligible. However, the detection delays should be taken into consideration [142, 171]. The reason for considering detection delays is because the use of the states that are received during delay interval may lead to a wrong control actions. To address the issue of the detection delays, based on a checkpointing protocol developed in [170], an RMPC scheme is developed to determine optimal cascade mitigation actions under the impacts of cyber-physical attacks and the attack detection delays. The strategy of the checkpointing protocol is illustrated in Fig. 5.6. Assume the attack detection method uses the latest received data of length N , i.e., from time $k-N+1$ to the current time k , to determine the attacks. If the detection method clears the examined data, the cascade mitigation actions can be determined based on the measured data at time k . However, if the examined data is found to be untrustworthy, this latest collected data of length N cannot be used anymore as it is hard to identify the exact attack injection time. To address this challenge, the trustworthy stored data at time instant $k - N$ are used to first recover the state at time k , and then proper control actions are identified. The stored correct system state at time instant $k - N$ is referred to as a checkpoint, that are prepared for system state

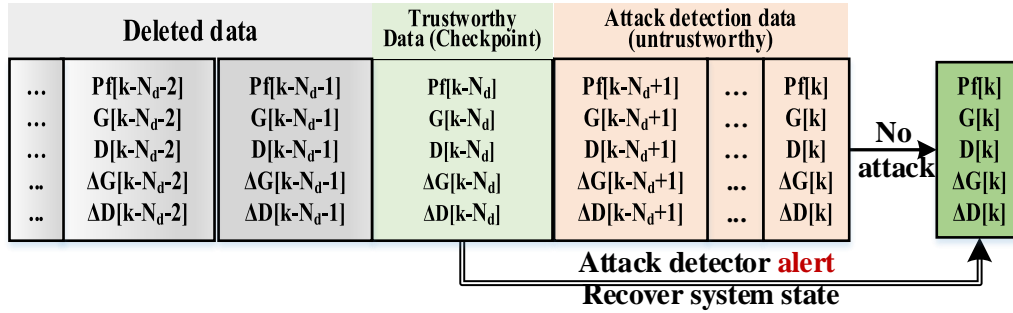


Figure 5.2: Checkpointing protocol illustration in handling attack detection delays

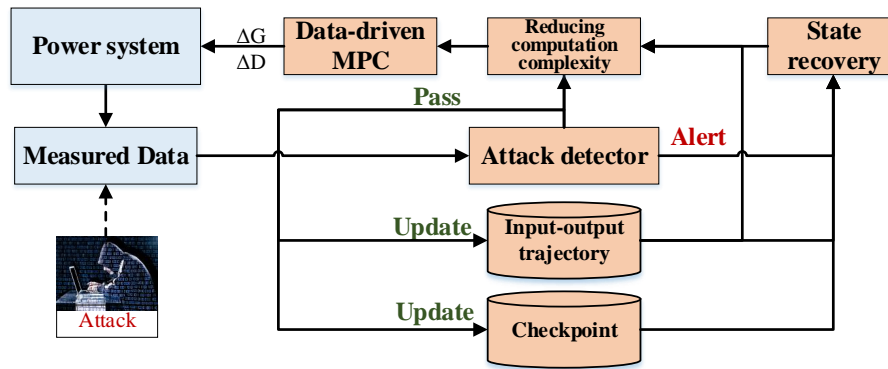


Figure 5.3: The RMPC architecture for mitigating cascades following a CCPA

recovery if an attack is detected. For the next time instant $k + 1$, the stored states at time $k - N + 1$ would be updated as a new checkpoint and the previous checkpoint at time $k - N$ is discarded. The architecture of the RMPC scheme is given in Fig. 5.3. Power system meters such as PMUs continue to measure the states of system variables, while the measured data is examined through an attack detector. If the measured data is deemed clear of any anomalies, the estimated system states will be sent to MPC controller and used to update the stored checkpoints. On the other hand, if the measured data is reported compromised, the state recovery process will be activated to use the historical data to predict the real-time system state, that in turn is sent to MPC to determine optimized control actions ΔP and ΔD . Note that the state recovery error is assumed to be smaller than the user defined maximum tolerance. If the recovery error is above the tolerance, the system may need to restore the tampered meters.

5.5 Recovery-based Data-driven MPC Model

5.5.1 Overview of the MPC Controller

The architecture of the developed recovery-based data-driven MPC (RD-MPC) is illustrated in Fig. 5.4. The specific type of attack considered in this chapter is an attack in which system states are altered. The system operating conditions are measured and transmitted to the control centers by PMUs or SCADA measurements. A malicious false data attack can potentially bypass bad data detection mechanisms and change the system states by altering relay status or tampering multiple measurements [172, 173], leading to wrong control actions that further aggravate system conditions. For example, during a FDIA, an attacker can disrupt the power flows (by means such as phase shifting transformer (PST) attacks), and at the same time inject false data to conceal the disruption. Thus, instead of directly using the measured data to determine the mitigative actions, the received data are first examined by an attack detector.

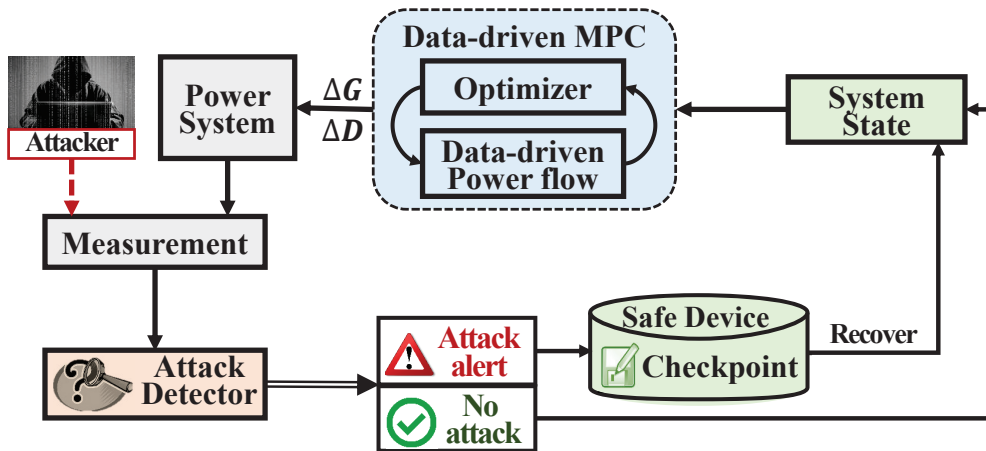


Figure 5.4: RD-MPC architecture for mitigating thermal overloads after a FDIA

Here, the attack detection method in Chapter 2 is leveraged, in which text mining methods are used to promptly identify tampered PMU measurements. Note that while only one type of attack detection is considered in this study, other attack detection methods can also

be deployed. However, attack detection is outside the scope of this study. Hence, a previously developed method is used. If the measurements do not exhibit malicious data, they are deemed safe to be used by the data-driven MPC controller to determine the optimal control actions. These correct measurements are also used to update the stored checkpoint, i.e., trustworthy historical data. However, if an attack detector indicates the existence of false data, the state recovery process will be triggered to use the stored checkpoint to estimate the actual system states. With these recovered system states, the data-driven MPC controller determines the corresponding optimal mitigative actions.

5.5.2 Construction of the Data-driven MPC Model

The data-driven MPC method utilizes the measurement-based power flow model that was described in the previous section. The aim is to identify optimal mitigative actions over a finite time horizon. Three model information are required to construct the data-driven controller: (1) the length of the prediction horizon N^{MPC} ; (2) the measured system input-output trajectories $\Delta Pinj_{[0, N-1]}$ and $Pf_{i, [0, N-1]}$; and (3) the system operating conditions represented by $\Delta Pinj(0)$, $Pf(0)$. The developed new data-driven MPC formulation for alleviating the overloads is formulated as,

$$\begin{aligned} \text{Minimize}_{\Delta G[k], \Delta D[k]} \quad & \sum_{k=0}^{N^{\text{MPC}}-1} (\Delta G^T[k] \cdot \mathbf{M} \cdot \Delta G[k] + \Delta D^T[k] \cdot \mathbf{Q} \cdot \Delta D[k]) \\ & + S^T[k] \cdot \mathbf{S}\mathbf{L} \cdot S[k] + \sum_{i=1}^{N_{\text{line}}} \mu \|\alpha_i\|_2^2 \end{aligned} \quad (5.13.1)$$

$$s.t. \quad \forall k \in \{0, 1, \dots, N^{\text{MPC}}\}, \quad \forall i \in \{\text{Lines}\}$$

$$\begin{bmatrix} \mathbf{H}_1^{\text{ini}}(\Delta Pinj) \\ \mathbf{H}_{N^{\text{MPC}}}^f(\Delta Pinj) \\ \mathbf{H}_1^{\text{ini}}(Pf_i) \\ \mathbf{H}_{N^{\text{MPC}}}^f(Pf_i) \end{bmatrix} \alpha_i = \begin{bmatrix} \Delta \overline{Pinj}(0) \\ \Delta \overline{Pinj}_{[1, N^{\text{MPC}}]} \\ \overline{Pf}_i(0) \\ \overline{Pf}_{i, [1, N^{\text{MPC}}]} \end{bmatrix} \quad (5.13.2)$$

$$|\overline{Pf}[k]| \leq Pf^{\max} + S[k] \quad (5.13.3)$$

$$|\overline{Pf}[N^{\text{MPC}}]| \leq Pf^{\max} \quad (5.13.4)$$

where $\overline{Pf}_{i, [1, N^{\text{MPC}}]}$ is the power flow of line i over the next predicted N^{MPC} time steps. The weight matrices $\mathbf{M} \in \mathbb{R}^{N_{\text{gen}} \times N_{\text{gen}}}$ and $\mathbf{Q} \in \mathbb{R}^{N_{\text{load}} \times N_{\text{load}}}$ penalize generation ramping and load shedding. The slack variable vector $S \in \mathbb{R}^{N_{\text{line}}}$ converts the power flow limits, i.e. $|\overline{Pf}[k]| \leq Pf^{\max}$, into soft constraints such that feasible solutions are obtained. The penalty weight matrix for the slack variable is $\mathbf{SL} \in \mathbb{R}^{N_{\text{line}} \times N_{\text{line}}}$. A transmission line can withstand a slight overload for a short time, which is the rational for using the slack variable S .

With the objective function of the MPC formulation in (5.13.1), the cost of generation ramping and load shedding is minimized. The data-driven power flow model of each transmission line is formulated in (5.13.2), and the four Hankel matrices explained in Lemma 2 are constructed with $N_{\text{initial}} = 1$ and $N_f = N^{\text{MPC}}$. The terminal constraint (5.13.4) guarantees that all line power flows below their limits at the end of the prediction horizon [174]. Other constraints of control actions, load and generators can be found in a previous study [175].

5.5.3 Reducing Computational Time

The data-driven MPC determines the optimal actions in multiple future time steps. As seen in (5.13), the developed controller considers a large set of variables (e.g., S , ΔG and ΔD for all the generators and controllable loads) and constraints. This drawback is addressed by decreasing the number of variables and constraints before employing the data-driven MPC. Two strategies are developed to achieve this goal: 1) effectively limiting the control actions, and 2) identifying a smaller set of controlled lines.

The first strategy is to reduce the control action space, i.e. generation ramping and load

shedding while keeping the ability to control the overloads intact. As indicated by other studies, only the actions taken on a subset of generators or loads have a significant impact on the power flow of a particular line [176]. This is also corroborated by the PTDF matrix, where some elements are zero or close to zero. Thus, it is suggested in the literature that the location of the sparse elements of the PTDF matrix can be used to reduce the number of control actions [162]. However, the PTDF matrix might be erroneous due to topology processor errors. This problem is addressed by developing an alternative method to using the PTDF matrix, i.e., the buses with reduced control actions are directly determined via system measurements. To specify the location of the controlled buses in a time more suitable for online applications, the formulation in (5.13) is simplified by setting N^{MPC} to one, which results in a more aggressive mitigation strategy that aims to alleviate overloads in only one time step. The objective is to reduce the number of control actions. Hence, the zero norm of the control vector is minimized, while characteristics such as the data-driven power flow model, and power flow limits are retained,

$$\underset{v}{\text{Minimize}} \quad v^T \mathbf{C}v + \rho_1 \|v\|_0 + \sum_{i=1}^{N_{\text{line}}} \rho_2 \|g_i\|_2^2 \quad (5.14.1)$$

$$s.t. \quad \forall i \in \{\text{Lines}\}$$

$$\begin{bmatrix} \mathbf{H}_{N_{\text{ini}}}^{\text{ini}}(\Delta P_{\text{inj}}) \\ \mathbf{H}_{N_{\text{ini}}}^{\text{ini}}(Pf_i) \\ \mathbf{H}_{N_f}^f(Pf_i) \end{bmatrix} g_i = \begin{bmatrix} v \\ Pf_i(0) \\ Pf_i(1) \end{bmatrix} \quad (5.14.2)$$

$$\|Pf(1)\| \leq Pf^{\text{max}} \quad (5.14.3)$$

$$\sum_{j=1}^{N_{\text{bus}}} v(j) = 0 \quad (5.14.4)$$

$$v_{\text{min}} \leq v \leq v_{\text{max}} \quad (5.14.5)$$

where $v \in \mathbb{R}^{N_{\text{bus}}}$ is the control action vector, and $\mathbf{C} \in \mathbb{R}^{N_{\text{bus}} \times N_{\text{bus}}}$ is the control action cost at each bus. The penalty parameter ρ_1 determines the number of the selected control

actions, and ρ_2 is the regularization parameter. The objective of (5.14.1) is to minimize the cost of control actions and the number of non-zero elements (via zero norm) in the control vector v . The formulation described in (5.14.1) is non-convex, due to existence of the l_0 -norm. Hence, the l_0 -norm is relaxed using a l_1 -norm, which yields a convex optimization problem,

$$\underset{v}{\text{Minimize}} \quad v^T C v + \rho_1 \|v\|_1 + \sum_{i=1}^{N_{\text{line}}} \rho_2 \|g_i\|_2^2. \quad (5.15)$$

An example solution to (5.14) for the IEEE 14-bus system is given in Fig. 5.5. It is seen that, instead of taking actions at all buses to control the power flow, the selected control actions are in vicinity of the regulated $line_{2-4}$. The control action reduction problem can be viewed as a simplified data-driven MPC where $N^{\text{MPC}} = 1$. The solution to this selection problem identifies the location and the number of the control actions as well as the approximate power flows after taking the control actions. With this approximate power flow, the second time reduction strategy is introduced to further reduce the computation time.

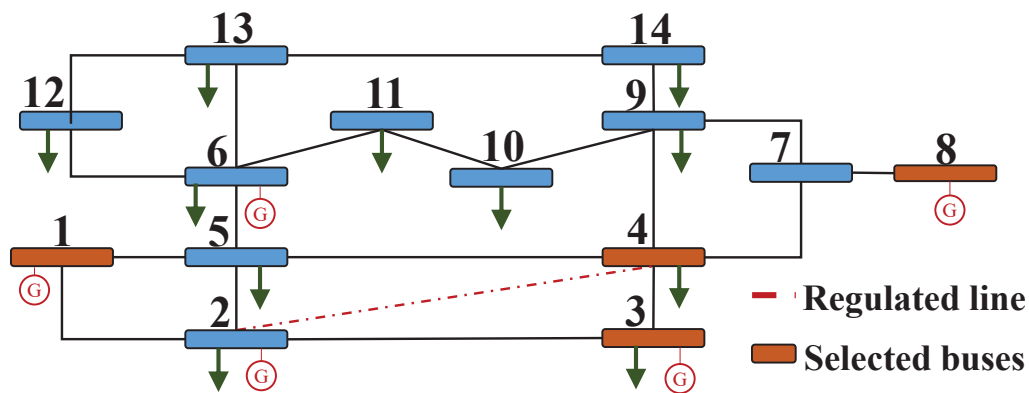


Figure 5.5: Example of a control action reduction for the IEEE 14-bus system. In order to regulate $line_{2-4}$, control actions are applied to buses 1, 3, 4 and 8. The parameters in (5.14) are $\rho_1 = 5$ and $\rho_2 = 0.1$.

The second strategy is to reduce the number of controlled lines in (5.13.2). The power flows of many lines are far from their maximum limits, and stay within their limits after taking control actions. Hence, excluding these lines from the MPC calculations would not affect the results of the data-driven MPC, while significantly improving the computation

time. The approximate line power flows, after applying the predicted control actions from (5.14), are used to determine the lines that can be eliminated. The rationale is that, if the line power flows are less than their limits after taking the aggressive mitigative actions from (5.14), these lines will also be within the limits when taking the moderate actions from (5.13). With the obtained power flows of all lines from (5.14), a line i with a power flow of smaller than $\gamma \cdot P f_i^{\max}$ can be eliminated when solving (5.13). The parameter γ is used to control the number of the selected lines. Larger γ can result in fewer controlled lines in (5.13), however, the unexpected overloads during MPC prediction maybe neglected.

5.5.4 Improving Attack-resiliency

Attack resiliency of the developed method can be improved by recovering the wrong states at time k from the stored trustworthy data at time $k - N_d$. By extending the state recovery method in [175] to the data-driven power flow model, the system states are rolled forward to the present states at time instant k from the past correct system states at time $k - N_d$. Specifically, given the stored control actions, i.e. ΔG and ΔD , during time $k - N_d$ to k , the corresponding power injection changes $\{\Delta P inj(t)\}_{t=k-N_d}^k$ can be computed. Hence, given the stored trustworthy system state at time $k - N_d$, i.e., $P f^{\text{trust}}(k - N_d)$, the power flow at time k can be estimated with the developed data-driven power flow model, that in turn can be used to amend the wrong states at time k .

The state recovery concept recovers the attacked system states. However, the FDIA detection delays are not taken into consideration. Conventional detection methods use a sequence of received data to detect malicious attacks. During the attack detection window, the received data sequence cannot yet be used to recover any wrong states, as their trustworthiness still needs to be verified. The duration of this data sequence within the detection window is referred to as a *detection delay*. When an attack is detected within

a detection window, it is impractical to determine at what point in time the attack is injected. Thus, utilizing the data within the detection window may not correctly recover the actual system states. To address the detection delays, the checkpointing protocol inspired by [170], and depicted in Fig. 5.6, is extended here. An attack detection methodology such as [136] determines if the received data between time instants $k - N_d + 1$ to k , referred to as *attack detection data*, are maliciously modified. The attack detection data are pending to be stored, and thus are not yet trustworthy. The trustworthy data at time $k - N_d$, is referred to as a *checkpoint*, and is securely stored. If any attacks are identified, the stored checkpoint would be used to recover the actual system states at time k . If no attacks are detected, the measured data at time k are used to identify mitigations, and replace the previously stored checkpoint. The *deleted data*, are the previously stored checkpoints that are no longer needed for the state recovery. Since the focus of this chapter is to develop an effective recovery protocol instead of an attack detection method, our previously developed attack detection model in [136] is utilized to detect the attack.

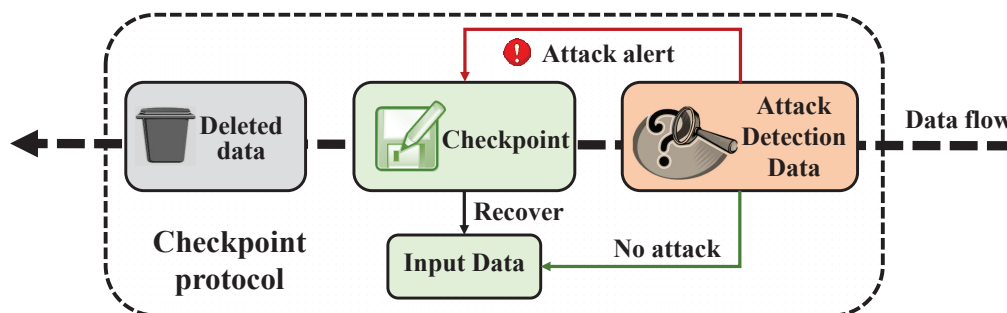


Figure 5.6: The Checkpointing protocol for addressing the attack detection delays

5.6 Simulation Results

The developed RD-MPC is tested on IEEE 30-bus, IEEE 118-bus and Illinois 200-bus systems. The performance of the data-driven MPC when mitigating overloads is first compared to the model-based MPC. The performance of the data-driven MPC with noisy measurements is also evaluated. Next, the RD-MPC is evaluated under a false data attack, to

demonstrate its resiliency to attacks. For these studies, it is assumed that all transmission lines are observed by PMUs.

5.6.1 Scenario Setup

1. PMU data generation

PMU time-series data, in particular, bus voltages and line currents, are simulated via MATPOWER toolbox [177]. At each sampling instant m , the load at bus i , with a load fluctuation of ω and a PMU measurement noise of ϵ_i , is denoted by,

$$D_i[m] = (1 + \omega)D_i^0 + \epsilon_i, \quad (5.16)$$

where $D_i[m]$ and D_i^0 refer to the load at bus i , at instant m and the nominal value, respectively. Given the load profile at each time instant, the voltage and generation at each bus, and the transmission line currents can be calculated via an AC power flow model. Thus, the bus active power injections at each time step can be obtained, and the power flow of lines can be obtained from PMU-measured bus voltages and line currents. Without loss of generality, and inspired by [165, 178], it is assumed that ω and ϵ follow a normal distribution with a zero mean and a standard deviation of 0.1 and 0.01, for ω and ϵ respectively. However, other distributions are expected to yield the same results sought in this chapter. The sampling frequency of the PMUs is 60 frames per second.

2. Attack Strategy

In this study, two different coordinated attacks are considered. To demonstrate the effectiveness of the RD-MPC against wrong model information, a coordinated attack that causes line overloads and fake outages (by PST physical attack and relay status attack), is simulated in three different test systems. These coordinated attacks are illustrated in Table 5.1. Fake outages are created by injecting fake relay data to change

the status of lines. The RD-MPC is further evaluated with another type of coordinated attack in IEEE 118-bus system. The physical attack results in the overloads on lines 22, 28, while the overload on line 22 is concealed by tampering the power flow and load measurements.

Table 5.1: Details of the coordinated attack scenarios

	Test System	Objective	Strategy
Relay attack	IEEE 30-bus	Fake outage of line 27	PST physical attack (lines 10, 22, 35 overloaded)
			relay attack
	IEEE 118-bus	Fake outage of line 28	PST physical attack (lines 22, 26 overloaded)
			relay attack
	Illinois 200-bus	Fake outage of line 54	PST physical attack (lines 27, 54 overloaded)
			relay attack
Measurement attack	IEEE 118-bus	Conceal overload line 22	PST physical attack (lines 22, 26 overloaded)
			Tamper measurements Pf_{22}, D_{16}, D_{17}

5.6.2 Model Parameter Setup

The matrices containing the weights of the objective function in formulations described in (5.13), namely, \mathbf{M} , \mathbf{Q} and \mathbf{SL} , are defined as $\mathbf{M} = \mathbf{I}_g \times \omega_g$, $\mathbf{Q} = \mathbf{I}_d \times \omega_d$ and $\mathbf{SL} = \mathbf{I}_s \times \omega_s$, where \mathbf{I}_g , \mathbf{I}_d and \mathbf{I}_s are identity matrices with proper dimensions. Since the ramping costs of the generators are not known, the cost curve parameter of the quadratic term, i.e. a_i , in the quadratic cost function ($\text{Cost}(G_i) = a_i G_i^2 + b_i G_i$) is used for setting generator ramping weights. The normalized weight for the i^{th} generator, i.e. $\omega_g(i)$ is defined as:

$\max\{0.1, \frac{a_i}{\max_i\{a_i\}}\}$ [161]. The value 0.1 is used to avoid the zero weight due to the fact that the a_i parameters of some generators may be zero. The ramp rate (MW/min) of the generators is modeled as 10% of their capacity [179]. If generator ramping is not enough to mitigate the overloads within the MPC time frame, load shedding will alleviate the remaining overloads as a last resort. Since load shedding is the last resort among all the control actions, the weights of ω_d are all set to 100. The weights of the slack variables S are set to 1000 to minimize the impact of the introduced soft constraints in (5.13.3). The regularization parameters μ in (5.13.1) and ρ_2 in (5.14.1) are set to 0.01 to reduce the impacts of the measurement noise [180].

Since the transmission lines can withstand thermal overloads for a short duration [181], the length of RD-MPC sampling time is set to 60 seconds. To match the sampling frequency of the RD-MPC with the generated PMU data and construct the historical input-output trajectory $\{\Delta Pin_{[0,N-1]}, Pf_{[0,N-1]}\}$, the voltage and current PMU measurements are sampled every 60 seconds from the simulated PMU time-series data. The length of the prediction horizon N^{MPC} is set to five minutes. The shorter the prediction time is, the faster the overloads are alleviated. However, the cost for the control actions increases.

The regularization parameter ρ_1 , used for controlling the sparsity of the control action vector v in (5.15), is set to one. Larger ρ_1 may lead to a more sparse control vector, i.e. fewer control actions. However, the obtained control actions may lead to infeasible RD-MPC solutions, i.e. fewer control actions are insufficient for mitigating the overloads. In addition to the control action selection strategy, the parameter γ from the second computation reduction method is set to 0.9 to avoid missing line overloads. When the line power flows, obtained from (5.14), are smaller than $0.9Pf^{\max}$, they are not considered when solving RD-MPC. Large values of γ result in fewer lines being selected. However, some overloads may be missed. With the aforementioned linear power flow assumptions in Section 5.3.2, and the fact that in real power systems $|I_{ij}| \approx |P_{ij}|$, the use of active power

for overload mitigation is justified [182].

5.6.3 RD-MPC against Wrong Topology

The developed data-driven MPC is evaluated on IEEE 30-bus, IEEE 118-bus and Illinois 200-bus systems. To satisfy the conditions of Lemma 2, i.e., $N \geq (N_{\text{bus}} + 1)N^{\text{MPC}} + N_{\text{bus}}$, the length of the historical input-output trajectory for these three systems are set to 500, 800, and 1300, respectively. Given the overload mitigation scenarios in Table 5.2, the computation reduction strategies are first implemented to limit the considered number of constraints and variables. The number of selected control actions and the regulated lines from the developed data-driven MPC for each test system are given in Table 5.2. Compared to the total numbers, the number of control actions and lines are significantly reduced.

Table 5.2: Thermal overload mitigation scenarios, the optimal number of control actions, and the selected lines for three systems

Test Systems	IEEE 30-bus			IEEE 118-bus		Illinois 200-bus	
	System			System		System	
Regulated Line	10	22	35	22	26	27	54
Actual Electric Current (pu)	0.325	0186	0.169	0.25	0.135	0.926	0.515
Desired Current (pu)	0.32	0.16	0.16	0.20	0.10	0.90	0.47
Number of selected controlled buses / total buses	8/30			29/118		69/200	
Number of selected lines / total lines	7/41			31/186		14/245	

Once the controllable generator and loads are specified and the number of considered lines are reduced, the data-driven MPC yields optimal control actions, which are generation ramping ΔG and load shedding ΔD . The data-driven MPC is compared to the conventional model-based MPC. Specifically, three different conditions are compared to demonstrate the effectiveness and robustness of the developed data-driven MPC: model-

based MPC with correct, and wrong models, and data-driven MPC with a wrong model. A wrong model can be caused by errors in the topology processor or incorrect relay status information.

The control results are illustrated in Figs. 5.7 to 5.9. For the IEEE 30-bus system, controlling the current of line 22 shows that the data-driven MPC can mitigate an overload on this line and maintain the current within its limit of 0.16 pu in 5 time steps, even with a wrong system model (the topology processor assumes line 27 is out of service by mistake). It is also found that the data-driven MPC achieves a comparable mitigation performance compared to the model-based MPC that uses a correct model. With an inaccurate system model, the model-based MPC fails to find appropriate control actions, as observed in Fig. 5.7. The control results for both IEEE 118-bus and Illinois 200-bus systems also demonstrate that the data-driven MPC effectively regulates the line current, even in the presence of wrong system model information. A model-based MPC with a wrong system model, may still successfully mitigate overloads (Fig. 5.9). The PTDF matrix of the wrong system model in Illinois 200-bus system shows that the line with wrong status information is not electrically close to the regulated line 54. Hence, the PTDF matrix does not change significantly with the wrong information. This example shows that the performance of the model-based MPC is dependent on the location of the wrong information, while the developed data-driven MPC performs well under all circumstances.

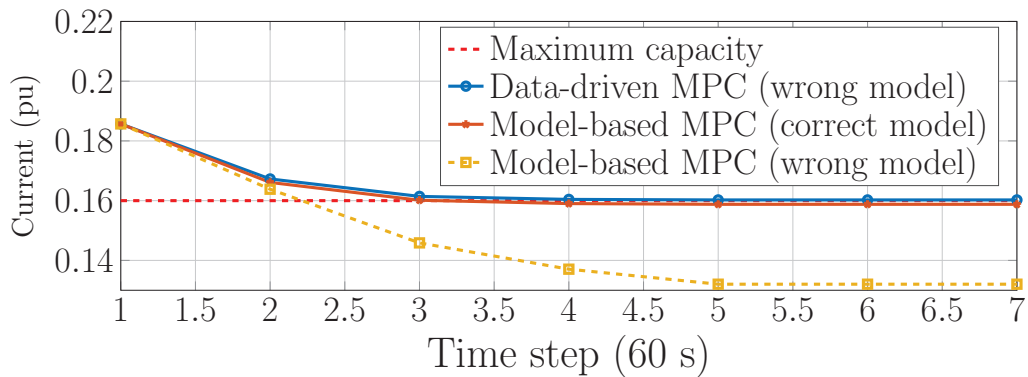


Figure 5.7: Performance comparison for mitigating an overload on line 22 in 30-bus system. Line 27 is incorrectly assumed outaged. A time step is 60s.

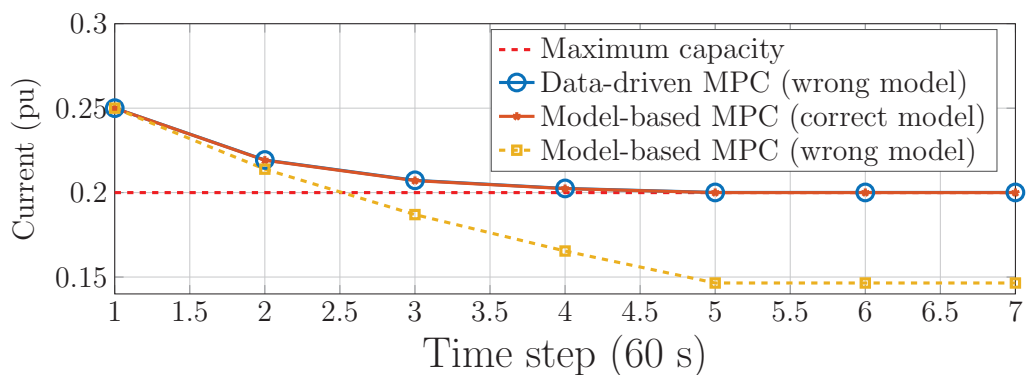


Figure 5.8: Mitigating an overload on line 22 in 118-bus system with a wrong system model, that incorrectly assumes lines 15 and 20 are outaged.

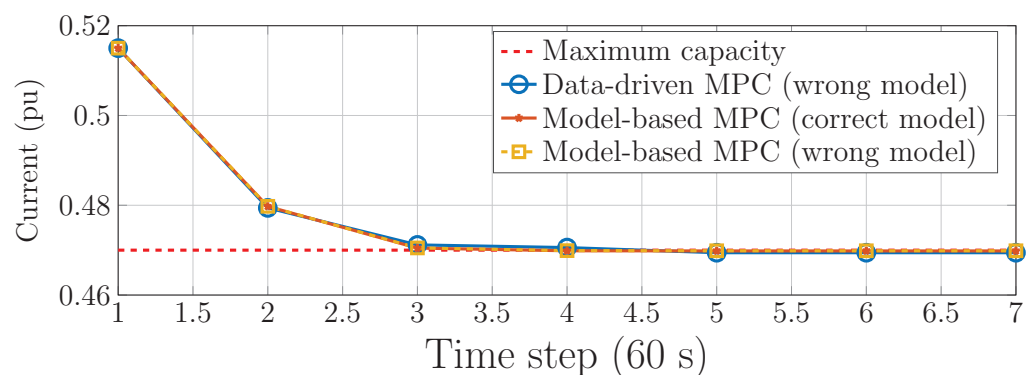


Figure 5.9: Performance comparison of the data-driven MPC in mitigating an overload on line 54 of the 200-bus system. Line 38 is assumed to be out of service in the wrong model.

5.6.4 RD-MPC against False Data

A malicious targeted FDIA intentionally tampers the measured data to bypass bad data detection and misinforms system operators of the real system operating conditions [144]. To improve the resiliency of the data-driven MPC against FDIA attacks, the RD-MPC that incorporates checkpointing protocol is developed, as explained in section 5.5.4.

Wrongly estimated system states may lead to incorrect control actions that further jeopardize system reliability. The developed RD-MPC is evaluated on the following FDIA scenario (given in Table 5.1): line 22 of the IEEE 118-bus system is overloaded by 0.05 pu at time instant 5, while the attackers change the measurements to hide the overload. The results of the RD-MPC and data-driven MPC under attacked data, and the data-driven MPC with correct data are given in Fig. 5.10. Under FDIA, the obtained control actions from RD-MPC successfully regulate currents in five time steps, while the data-driven MPC fails to mitigate the overload on line 22. With state recovery, RD-MPC achieves similar control performance as the data-driven MPC with correct data.

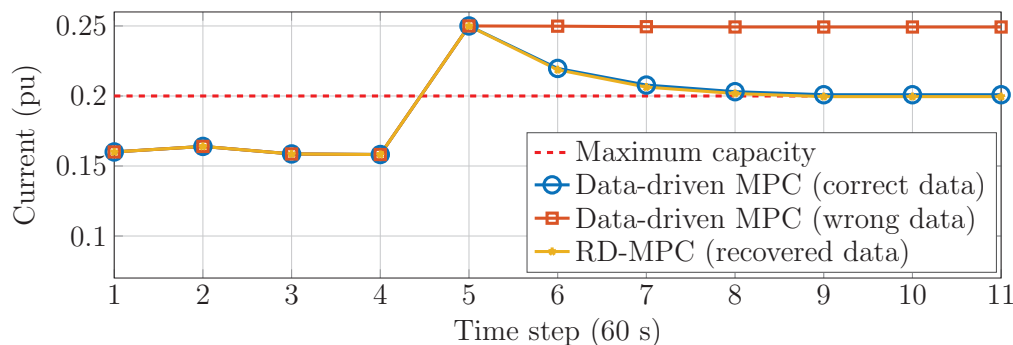


Figure 5.10: Performance comparison after an overload on line 22 in 118-bus system. A successful attack hides the overload. Mitigation provided in Table 5.2.

5.7 Discussion

5.7.1 Robustness to Noise

To evaluate the robustness of the data-driven MPC against noise, its overload mitigation capability for IEEE 118-bus system, under different signal to noise ratios (SNR) of 60dB, 70dB, 80dB, and 90dB, are compared. Following the same approach taken in [137], and [138], Gaussian noise is added to the simulated PMU data. The results presented in Fig. 5.11 show that even with different noise levels, the data-driven MPC can still effectively regulate power flows. The robustness of the data-driven MPC against noise is due to the introduced regularized squared method in (5.10) that leads to a more sparse α . Thus, when multiplied by the Hankel matrix, that includes noisy data, the impact of noise is reduced. Although a recent study in [183] observes that the PMU noise is non-Gaussian, the impact of other types of noises can still be minimized by a sparse α vector.

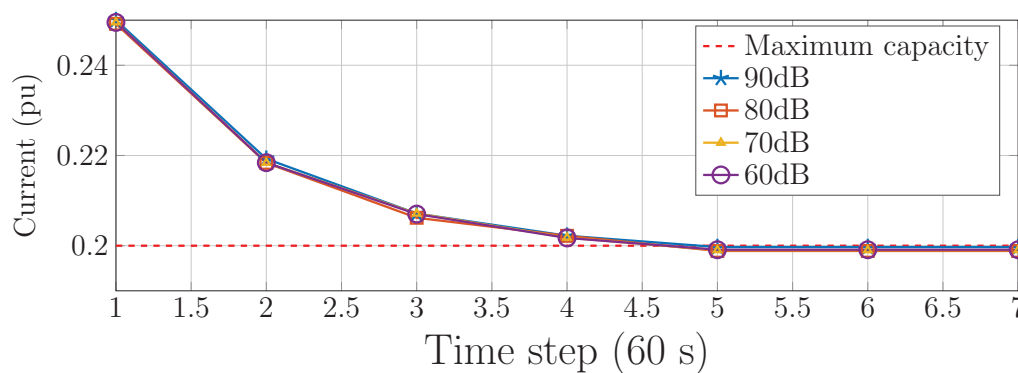


Figure 5.11: Evaluation of the robustness of the data-driven MPC against noisy data in mitigating the overload on line 22 in 118-bus system. Four different SNR levels are considered.

5.7.2 Computation Time

The computation time of the data-driven MPC for Illinois 200-bus system with different sparse regulation parameters ρ_1 , and line selection parameter γ is compared in Table 5.3.

With these strategies, the computation time drops to 39.9 s from 71.54 s, given $\gamma = 0.9$ and $\rho_1 = 1$. A slight reduction in γ increases the computation time, due to an increase in the number of the selected lines. Increasing ρ_1 reduces the number of control actions, and thus slightly decreases the computation time. However, large ρ_1 may cause infeasible solutions for the data-driven MPC. With the described time reduction strategies, the total time for the data-driven MPC, including solver computation and optimization model construction time, is 0.9s, 9.7s, and 39s for the IEEE 30-bus, IEEE 118-bus and Illinois 200-bus systems, respectively. The solver computation times are 0.47s, 0.58s and 0.6s, respectively. The computation times are less than a minute, while the model construction time is the most time consuming step. However, the optimization model only needs to be constructed once. Thus, the data-driven MPC can potentially be used to promptly mitigate thermal overloads. For a power system with thousands of buses, a decentralized computation approach can further reduce the time.

Table 5.3: Computation time of the data-driven MPC for Illinois 200-bus system. The mitigation scenario is given in Table 5.2.

		# buses	# lines	Time (s)
0	0	200	245	71.54
0.9	1	69	14	39.9
0.8	1	69	19	40.7
0.9	1.5	60	14	39.6

5.7.3 Limitation and Improvement

Here, the limitations of the developed MPC method are discussed, and possible solutions are introduced. Line outage attacks, which result in a change in system topology, are not considered in this chapter. A topology change requires reconstruction of the data-driven power flow model with new system data after line outages. The time for such model reconstruction will be more than a minute for a large power system, which is not desired

for the prompt overload mitigation considered. To overcome the aforementioned limitation, two strategies can be taken: (1) increasing the sampling frequency of the PMUs; (2) utilizing a decentralized MPC method to find the optimal actions. The rationale for increasing the sampling frequency is that the required number of PMU data for constructing the power flow model can be obtained in a shorter time. A decentralized MPC method further improves the computation time, by reducing the dimension of the power flow model. The decentralized approach divides the system into multiple sub-systems, controlled by a separate MPC controller [184].

Another limitation is that the developed RD-MPC requires the grid to be fully observable by PMUs. Loss of observability would result in an inaccurate data-driven power flow model, and thus cannot correctly mitigate overloads. To ensure the effectiveness of the RD-MPC method, redundant PMUs should be deployed to guarantee the observability of the grid, if any PMUs become unavailable.

5.8 Conclusion

This chapter developed an attack-resilient control method (RD-MPC) for mitigation of the thermal overloads under the threats of targeted FDIA. While conventional MPC control methods rely on DC power flow model to identify proper mitigative actions, RD-MPC estimates a data-driven power flow model through measured input-output PMU data, and thus avoids the adverse impacts of a wrong system model. To address the possibility of a FDIA on PMU data, a checkpointing protocol is adopted and extended to recover the actual system states from the past trustworthy data, if the measured data is believed to be compromised. Two strategies were introduced to accelerate the RD-MPC computation time, and make it applicable to practical applications. Simulation results on IEEE 30-bus, IEEE 118-bus, and Illinois 200-bus systems demonstrated that, unlike the conventional MPC, the RD-MPC can still use the measured data to identify the optimal control actions

and eliminate the thermal overloads, even in presence of wrong system model or noisy measurements.

Chapter 6

Summary and and Future Work

In this chapter, the main contributions of this dissertation, i.e., grid event detection, impact assessment, and mitigations, are summarized. The future directions of the research are also discussed here.

6.1 Summary

This dissertation focuses on improving grid reliability with PMUs in three aspects: grid event detection, impact assessment in both steady and dynamic states, and event mitigation.

In Chapter 2, a new data-driven approach is developed to rapidly and accurately inform system operators of the types of system events using PMU voltage and current data. The conventional event classification methodologies cannot distinguish counterfeit events that are generated by the attackers. However, in the developed event classification approach, counterfeit events can be labeled as a fake event. Hence, this methodology enables a more credible knowledge of the system events. To promptly and accurately process large-scale PMU data and extract event oscillation patterns, an effective data processing method is developed. Rather than only examining a measurement with the largest variation during an event, the developed data-driven approach analyzes the multivariate PMU times series data gathered at local PDC such that the correlation among PMU time series data can be determined and utilized to identify counterfeit events. Compared to the baseline methods such as Principal Component Analysis (PCA), wavelet, shapelet and Dshapelet, the devel-

oped data driven method achieves a higher classification accuracy on both clean and noisy PMU data. Furthermore, using substation measurements available at each local PDC, this chapter introduces a scalable approach to event classification, that is independent of the power system size. Instead of processing all PMU data as a whole, the proposed scalable approach shifts the event classification task from system level to the substation level. Therefore, the developed methodology can be applied to different power network sizes, a significant contribution over the state-of-the-art methods.

In Chapter 3, to evaluate the impact of the real system event and promptly predict potential cascading failures in steady state, a dynamic cascading failure model, that is adaptive to changing operating conditions, is developed. In this failure model, Bayesian framework is utilized to estimate power grid component interactions for *a specific operating condition*, resulting in more accurate estimation of the interactions for near real-time operations. Additionally, to predict the propagation of failures and predict the failure probability for each system component at each point in time, a propagation tree method is developed. This information can help initiate actions on the components with high failure probability, e.g. relay blocking, so that the propagation of cascades is prevented. Hence, the operators could have more time to identify optimal mitigative actions. Additionally, the developed model can forecast impending failures in a significantly shorter time than the DC/AC-based cascading failure models, and thus is more useful for power system monitoring and control.

In Chapter 4, the impact assessment of the grid event in transient state is considered. To accurately predict the transient status after a disturbance, a HGAN-based TSA model is developed. In this HGAN model, only a single sample of post-contingency PMU voltage data is needed. Hence, TSA can be performed in a relatively short period of time. Additionally, to retain the spatial and temporal features of the multivariate PMU time series data and improve TSA accuracy, the generative and adversarial networks are designed

in a specific manner. In case studies, the ability of the developed model in predicting a sequence of the transient data is demonstrated. In comparison to the conventional machine learning-based TSA methods that deploy decision trees, SVM, LSTM, or GRU, the developed model achieves better performance in terms of prediction accuracy and speed.

In Chapter 5, another aspect of enhancing grid reliability, i.e., event mitigation, is presented. To efficiently alleviate thermal overloads under false data injection attack, a data-driven controller, based on MPC approach, is developed. In this data-driven controller, a new measurement-based power flow is used to replace the approximate model-based DC power flow. In contrast to parametric data-driven methods that require calculating system parameters (such as ISF), the developed data-driven method is non-parametric, and thus avoids the impact of wrong system model information. To make the model robust to the false data attack, the concept of state recovery is leveraged to ensure the validity of the measurements used for identifying the proper mitigations. Furthermore, to make the control methodology more suitable for online applications, two strategies are developed to simplify the control action space while retaining feasible solutions.

6.2 Future Research Directions

6.2.1 Cascading Failure Analysis in both Steady and Dynamic States

In Chapter 3, a dynamic cascading failure model is developed to predict the cascade sequence and identify critical system components regarding steady state. Similar to the state-of-the-art cascading failure models, all these failure models neglect either steady or transient states to study the sequence of cascades. However, in reality, the propagation of the cascades is affected by factors in both steady and transient states, e.g., line overloads and unstable transients. To precisely study the cascades, combining both steady and transient states in analyzing the propagation of cascades is one of our future work.

6.2.2 Distributed Data-driven MPC for Alleviating Thermal Overloads

In Chapter 5, a centralized data-driven MPC model is developed to alleviate thermal overloads. When an overload occurs in power grids, all PMU measurements will be sent to the control center to help determine proper mitigative actions. As the size of the grid network increases, the time of identifying appropriate control actions by solving the MPC optimization problem will be significantly increased, and reducing the applicability for online applications. To address the issue of computation time, splitting the centralized MPC task into each local agent would dramatically decrease the computational complexity of the task. Hence, developing a distributed data-driven MPC model is considered as a future work.

6.2.3 Transient Event Sequence Prediction

In Chapter 4, a HGAN-based TSA approach is developed to accurately and promptly determine the transient status, i.e., stable or unstable, with only one sample of PMU voltage measurement. In this work, the main objective is to perform a classification task, although it is shown that the developed HGAN model has the ability to predict the sequence of the transient. In transient analysis, having the knowledge of the trajectory of transients is more critical, as the transient stability margin, i.e., the degree of power grid generators to remain synchronism, can be obtained. Hence, with only one sample of the post-fault PMU data, it is interesting to further improve the developed HGAN-based TSA model to predict the transient sequence.

Nomenclature

ΔD	Load shedding vector
ΔG	Generation ramping vector
$\Delta Pin,j(k)$	A column vector of active power injection change at time k
$\Delta Pin,j_{[0,N-1]}$	A stacked vector of active power injection change from time 0 to time N
δ_i	Rotor angle of machine i
γ, ρ_1	Regularization parameter for the number of selected lines, and the buses
Ψ	Power transfer distribution factor (PTDF) matrix
\mathbf{A}	Edge-to-node incidence matrix
$\mathbf{M}, \mathbf{Q}, \mathbf{SL}$	Weight matrix of generation ramping, load shedding, and the slack variable S
μ, ρ_2	Regularization parameter for the data-driven power flow model
ω_i	Rotor speed of machine i
σ	Sigmoid activation function
B_{ij}, G_{ij}	Conductance and susceptance in the i th row and j th column of admittance matrix \mathbf{Y}
D, G	Discriminator and Generator of the GAN model
D_i	Damping constant of generator i

E_i	Internal bus voltage of machine i
h_{t-1}	Output from the previous GRU
h_t	Final GRU output
h'_t	The current memory unit of GRU
M_i	Angular momentum of machine i
N^{MPC}, N_d	Length of the MPC prediction horizon and the attack detection horizon
N_{ini}, N_f	Length of the initial and future system input-output, respectively
P_i^m, P_i^e	Mechanical and electrical power input of the machine i
Pf^{max}	Line power flow limits
$Pf_{i, [0, N-1]}$	A column vector of active power flow of line i from time 0 to time N
$Pf_i(k)$	The active power flow of line i at time k
Pin_j	Power injections at all buses
r_t	The reset gate output of GRU
\tanh	Hyperbolic tangent activation function
$v_{\text{max}}, v_{\text{min}}$	Vectors of the maximal and minimal limits of the control actions
x_t	Input data to GRU
z_t	The update gate output of GRU
ϵ	PMU measurement noise

ω	Load fluctuation vector
ini, f	Abbreviations for initial and future system input/output

References

- [1] Y. Liu, M. K. Reiter, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” in *Proceedings of the 16th ACM conf. on Computer and communications security*, vol. 14. New York, USA: ACM Press, 5 2009, p. 21.
- [2] X. Liu and Z. Li, “Local load redistribution attacks in power systems with incomplete network information,” *IEEE Trans. on Smart Grid*, vol. 5, no. 4, pp. 1665–1676, 2014.
- [3] J. Minkel, “The 2003 northeast blackout—five years later,” *Scientific American*, vol. 13, 2008.
- [4] D. Wu, X. Zheng, Y. Xu, D. Olsen, B. Xia, C. Singh, and L. Xie, “An open-source model for simulation and corrective measure assessment of the 2021 texas power outage,” *arXiv preprint arXiv:2104.04146*, 2021.
- [5] A. Cherepanov and R. Lipovsky., “Industroyer: Biggest threat to industrial control systems since Stuxnet,” 2017. [Online]. Available: WeLiveSecurity,ESET12(2017).
- [6] G. Liang, S. R. Weller, J. Zhao, F. Luo, and Z. Y. Dong, “The 2015 Ukraine Blackout: Implications for False Data Injection Attacks,” *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3317–3318, 7 2017.
- [7] M. S. Thomas and J. D. McDonald, *Power system SCADA and smart grids*. CRC press, 2017.
- [8] S. Brahma, R. Kavasseri, H. Cao, N. Chaudhuri, T. Alexopoulos, and Y. Cui, “Real-

- time identification of dynamic events in power systems using pmu data, and potential applications—models, promises, and challenges,” *IEEE Trans. on Power Delivery*, vol. 32, no. 1, pp. 294–301, 2017.
- [9] S. Bruno, M. Benedictis, and M. Scala, “Taking the pulse of Power Systems: Monitoring Oscillations by Wavelet Analysis and Wide Area Measurement System,” in *2006 IEEE PES Power Systems Conference and Exposition*. IEEE, 2006, pp. 436–443.
- [10] S.-W. Sohn, A. J. Allen, S. Kulkarni, W. M. Grady, and S. Santoso, “Event detection method for the PMUs synchrophasor data,” in *2012 IEEE Power Electronics and Machines in Wind Applications*. IEEE, jul 2012, pp. 1–7.
- [11] A. Messina, V. Vittal, D. Ruiz-Vega, and G. Enriquez-Harper, “Interpretation and Visualization of Wide-Area PMU Measurements Using Hilbert Analysis,” *IEEE Trans. on Power Systems*, vol. 21, no. 4, pp. 1763–1771, nov 2006.
- [12] L. Fan, R. Kavasseri, Z. Miao, D. Osborn, and T. Bilke, “Identification of system wide disturbances using synchronized phasor data and ellipsoid method,” in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE, jul 2008, pp. 1–10.
- [13] Jian Ma, Y. V. Makarov, C. H. Miller, and T. B. Nguyen, “Use multi-dimensional ellipsoid to monitor dynamic behavior of power systems based on PMU measurement,” in *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE, jul 2008, pp. 1–8.
- [14] O. P. Dahal, S. M. Brahma, and H. Cao, “Comprehensive Clustering of Disturbance Events Recorded by Phasor Measurement Units,” *IEEE Trans. on Power Delivery*, vol. 29, no. 3, pp. 1390–1397, jun 2014.

- [15] S. S. Negi, N. Kishor, K. Uhlen, and R. Negi, "Event Detection and Its Signal Characterization in PMU Data Stream," *IEEE Trans. on Industrial Informatics*, vol. 13, no. 6, pp. 3108–3118, dec 2017.
- [16] M. Biswal, Y. Hao, P. Chen, S. Brahma, H. Cao, and P. De Leon, "Signal features for classification of power system disturbances using PMU data," in *19th Power Systems Computation Conference, PSCC 2016*, 2016.
- [17] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 149–182, jan 2011.
- [18] P. Senin and S. Malinchik, "SAX-VSM: Interpretable Time Series Classification Using SAX and Vector Space Model," in *2013 IEEE 13th International Conference on Data Mining*, Dec. 2013, pp. 1175–1180.
- [19] E. Keogh, L. Wei, X. Xi, M. Vlachos, S.-H. Lee, and P. Protopapas, "Supporting exact indexing of arbitrarily rotated shapes and periodic time series under Euclidean and warping distance measures," *The VLDB Journal*, vol. 18, no. 3, pp. 611–630, jun 2009.
- [20] E. Keogh, "Data mining and machine learning in time series databases," *Tutorial in SIGKDD 2004*, 2004.
- [21] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining and Knowledge Discovery*, vol. 15, no. 2, pp. 107–144, aug 2007.
- [22] J. Landford, R. Meier, R. Barella, S. Wallace, X. Zhao, E. Cotilla-Sanchez, and R. B. Bass, "Fast Sequence Component Analysis for Attack Detection in Smart

- Grid,” *2016 5th International Conference on Smart Cities and Green ICT Systems*, pp. 225–232, 2016.
- [23] S. Sridhar, A. Hahn, and M. Govindarasu, “Cyber–Physical System Security for the Electric Power Grid,” *Proceedings of the IEEE*, vol. 100, no. 1, pp. 210–224, jan 2012.
- [24] S. Basumallik, R. Ma, and S. Eftekharnjad, “Packet-data anomaly detection in PMU-based state estimator using convolutional neural network,” *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 690–702, may 2019.
- [25] X. Jiang, J. Zhang, B. J. Harding, J. J. Makela, and A. D. Dominguez-Garcia, “Spoofing GPS Receiver Clock Offset of Phasor Measurement Units,” *IEEE Trans. on Power Systems*, vol. 28(3), pp. 3253–3262, Aug. 2013.
- [26] V. Kekatos, G. B. Giannakis, and R. Baldick, “Grid topology identification using electricity prices,” in *2014 IEEE PES General Meeting — Conference & Exposition*. IEEE, jul 2014, pp. 1–5.
- [27] M. A. Rahman and H. Mohsenian-Rad, “False data injection attacks with incomplete information against smart power grids,” *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 3153–3158, 2012.
- [28] M. Esmalifalak, H. Nguyen, R. Zheng, and Zhu Han, “Stealth false data injection using independent component analysis in smart grid,” in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2011, pp. 244–248.
- [29] S. Basumallik, S. Eftekharnjad, N. Davis, and B. K. Johnson, “Impact of false data injection attacks on PMU-based state estimation,” in *2017 North American Power Symposium (NAPS)*. IEEE, sep 2017, pp. 1–6.

- [30] J. Jiang, X. Zhao, S. Wallace, E. Cotilla-Sanchez, and R. Bass, "Mining PMU Data Streams to Improve Electric Power System Resilience," in *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies - BDCAT '17*. New York, New York, USA: ACM Press, 2017, pp. 95–102.
- [31] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, "A Novel Data Analytical Approach for False Data Injection Cyber-Physical Attack Mitigation in Smart Grids," *IEEE Access*, vol. 5, pp. 26 022–26 033, 2017.
- [32] C. HUANG, F. LI, D. ZHOU, J. GUO, Z. PAN, Y. LIU, and Y. LIU, "Data quality issues for synchrophasor applications Part I: a review," *Journal of Modern Power Systems and Clean Energy*, vol. 4, no. 3, pp. 342–352, jul 2016.
- [33] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases," *Knowledge and Information Systems*, vol. 3 (3), pp. 263–286, Aug. 2001.
- [34] J. Lin, R. Khade, and Y. Li, "Rotation-invariant similarity in time series using bag-of-patterns representation," *Journal of Intelligent Information Systems*, vol. 39, no. 2, pp. 287–315, oct 2012.
- [35] M. G. Baydogan, G. Runger, and E. Tuv, "A Bag-of-Features Framework to Classify Time Series," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2796–2802, nov 2013.
- [36] P. Ordonez, T. Armstrong, T. Oates, and J. Fackler, "Using Modified Multivariate Bag-of-Words Models to Classify Physiological Data," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, dec 2011, pp. 534–539.

- [37] Y. Xiang, L. Wang, and N. Liu, “Coordinated attacks on electric power systems in a cyber-physical environment,” *Electric Power Systems Research*, vol. 149, pp. 156–168, 2017.
- [38] R. Ma and S. Eftekhariad, “Critical PMU Measurement Identification Based on Analytic Hierarchy Process,” in *IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, Apr. 2018, pp. 1–5.
- [39] North American Electric Reliability Corporation (NERC), “PRC-002-2-Disturbance Monitoring and Reporting Requirements.”
- [40] PowerTech Labs Inc, “TSAT Transient Security Assessment Tool,” 2007.
- [41] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning : data mining, inference, and prediction*.
- [42] L. Xie, Y. Chen, and P. R. Kumar, “Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis,” *IEEE Trans. on Power Systems*, vol. 29, no. 6, pp. 2784–2794, 2014.
- [43] P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen, “Discrete wavelet transform-based time series analysis and mining,” *ACM Computing Surveys*, vol. 43, no. 2, pp. 1–37, 2011.
- [44] C. Bishop and N. M. Nasrabadi, “Pattern Recognition and Machine Learning,” *Journal of Electronic Imaging*, vol. 16, no. 4, 2007.
- [45] S.-I. Lim, C.-C. Liu, S.-J. Lee, M.-S. Choi, and S.-J. Rim, “Blocking of zone 3 relays to prevent cascaded events,” *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 747–754, 2008.

- [46] I. Dobson, B. Carreras, V. Lynch, and D. Newman, “An initial model for complex dynamics in electric power system blackouts,” in *hicss*. Citeseer, 2001, p. 2017.
- [47] S. Mei, F. He, X. Zhang, and D. Xia, “An improved opa model and the evaluation of blackout risk,” *Automation of Electric Power Systems*, vol. 32, no. 13, pp. 1–5, 2008.
- [48] S. Mei, Y. Ni, G. Wang, and S. Wu, “A study of self-organized criticality of power system under cascading failures based on ac-opf with voltage stability margin,” *IEEE Trans. on Power Systems*, vol. 23, no. 4, pp. 1719–1726, 2008.
- [49] D. P. Nedic, I. Dobson, D. S. Kirschen, B. A. Carreras, and V. E. Lynch, “Criticality in a cascading failure blackout model,” *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 9, pp. 627–633, 2006.
- [50] J. Chen, J. S. Thorp, and I. Dobson, “Cascading dynamics and mitigation assessment in power system disturbances via a hidden failure model,” *International Journal of Electrical Power & Energy Systems*, vol. 27, no. 4, pp. 318–326, 2005.
- [51] P. Crucitti, V. Latora, and M. Marchiori, “A topological analysis of the italian electric power grid,” *Physica A: Statistical mechanics and its applications*, vol. 338, no. 1-2, pp. 92–97, 2004.
- [52] P. Walley, “Towards a unified theory of imprecise probability,” *International Journal of Approximate Reasoning*, vol. 24, no. 2-3, pp. 125–148, 2000.
- [53] C. D. Brummitt, R. M. D’Souza, and E. A. Leicht, “Suppressing cascades of load in interdependent networks,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 12, pp. E680–E689, 2012.

- [54] D. J. Watts, “A simple model of global cascades on random networks,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5766–5771, 2002.
- [55] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, “Catastrophic cascade of failures in interdependent networks,” *Nature*, vol. 464, no. 7291, p. 1025, 2010.
- [56] I. Dobson, B. A. Carreras, and D. E. Newman, “A loading-dependent model of probabilistic cascading failure,” *Probability in the Engineering and Informational Sciences*, vol. 19, no. 1, pp. 15–32, 2005.
- [57] I. Dobson, B. A. Carreras, and D. E. Newman, “Branching process models for the exponentially increasing portions of cascading failure blackouts,” in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*. IEEE, 2005, pp. 64a–64a.
- [58] R. Rocchetta, E. Zio, and E. Patelli, “A power-flow emulator approach for resilience assessment of repairable power grids subject to weather-induced failures and data deficiency,” *Applied Energy*, vol. 210, pp. 339–350, 2018.
- [59] B. Schäfer, D. Witthaut, M. Timme, and V. Latora, “Dynamically induced cascading failures in power grids,” *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.
- [60] F. Cadini, G. L. Agliardi, and E. Zio, “A modeling and simulation framework for the reliability/availability assessment of a power transmission grid subject to cascading failures under extreme weather conditions,” *Applied energy*, vol. 185, pp. 267–279, 2017.
- [61] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

- [62] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, “Complex networks: Structure and dynamics,” *Physics reports*, vol. 424, no. 4-5, pp. 175–308, 2006.
- [63] S. H. Strogatz, “Exploring complex networks,” *nature*, vol. 410, no. 6825, p. 268, 2001.
- [64] I. Dobson, B. A. Carreras, D. E. Newman, and J. M. Reynolds-Barredo, “Obtaining statistics of cascading line outages spreading in an electric transmission network from standard utility data,” *IEEE Trans. on Power Systems*, vol. 31, no. 6, pp. 4831–4841, 2016.
- [65] P. Hines, E. Cotilla-Sanchez, and S. Blumsack, “Do topological models provide good information about electricity infrastructure vulnerability?” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 20, no. 3, p. 033122, 2010.
- [66] P. D. Hines, I. Dobson, and P. Rezaei, “Cascading power outages propagate locally in an influence graph that is not the actual grid topology,” *IEEE Trans. on Power Sys.*, vol. 32(2), pp. 958–967, 2016.
- [67] K. Zhou, I. Dobson, Z. Wang, A. Roitershtein, and A. P. Ghosh, “A markovian influence graph formed from utility line outage data to mitigate cascading,” *arXiv preprint arXiv:1902.00686*, 2019.
- [68] J. Qi, K. Sun, and S. Mei, “An interaction model for simulation and mitigation of cascading failures,” *IEEE Trans. on Power Systems*, vol. 30, no. 2, pp. 804–819, 2014.
- [69] J. Qi, J. Wang, and K. Sun, “Efficient estimation of component interactions for cascading failure analysis by em algorithm,” *IEEE Trans. on Power Systems*, vol. 33, no. 3, pp. 3153–3161, 2017.

- [70] W. Ju, K. Sun, and J. Qi, "Multi-layer interaction graph for analysis and mitigation of cascading outages," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7(2), pp. 239–249, 2017.
- [71] Z. Zhang, S. Huang, Y. Chen, S. Mei, R. Yao, and K. Sun, "An online search method for representative risky fault chains based on reinforcement learning and knowledge transfer," *IEEE Transactions on Power Systems*, 2019.
- [72] R. Baldick, B. Chowdhury, I. Dobson, Z. Dong, B. Gou, D. Hawkins, H. Huang, M. Joungh, D. Kirschen, F. Li *et al.*, "Initial review of methods for cascading failure analysis in electric power transmission systems ieeepes task force on understanding, prediction, mitigation and restoration of cascading failures," in *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*. IEEE, 2008, pp. 1–8.
- [73] B. A. Carreras, J. M. Reynolds-Barredo, I. Dobson, and D. E. Newman, "Validating the opa cascading blackout model on a 19402 bus transmission network with both mesh and tree structures," 2019.
- [74] R. Christie and I. Dabbagchi, "30 bus power flow test case," 1993.
- [75] E. G. T. C. Respository. (2017) Illinois 200-bus system: Activsg200. [Online]. Available: <https://electricgrids.engr.tamu.edu/electric-grid-test-cases/activsg200/>
- [76] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*. Chapman and Hall/CRC, 2013.
- [77] M. Vaiman, K. Bell, Y. Chen, B. Chowdhury, I. Dobson, P. Hines, M. Papic, S. Miller, and P. Zhang, "Risk assessment of cascading outages: Methodologies and challenges," *IEEE Trans. on Power Systems*, vol. 27, no. 2, p. 631, 2012.

- [78] A. N. Philippou, C. Georghiou, and G. N. Philippou, "A generalized geometric distribution and some of its properties," *Statistics & Probability Letters*, vol. 1, no. 4, pp. 171–175, 1983.
- [79] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Trans. on power sys.*, vol. 32(4), pp. 3258–3265, 2016.
- [80] P. Jaccard, "Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines," *Bull Soc Vaudoise Sci Nat*, vol. 37, pp. 241–272, 1901.
- [81] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.
- [82] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.
- [83] N. Craswell, "Precision at n," *Encyclopedia of database systems*, pp. 2127–2128, 2009.
- [84] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.
- [85] A. Agresti, *Analysis of ordinal categorical data*. John Wiley & Sons, 2010, vol. 656.
- [86] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808*, 2018.
- [87] Q. Chen and J. D. McCalley, "Identifying high risk nk contingencies for online security assessment," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 823–834, 2005.

- [88] P. Kaplunovich and K. Turitsyn, “Fast and reliable screening of n-2 contingencies,” *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4243–4252, 2016.
- [89] D. Bienstock and A. Verma, “The nk problem in power grids: New models, formulations, and numerical experiments,” *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2352–2380, 2010.
- [90] V. Donde, V. López, B. Lesieutre, A. Pinar, C. Yang, and J. Meza, “Severe multiple contingency screening in electric power systems,” *IEEE Transactions on Power Systems*, vol. 23, no. 2, pp. 406–417, 2008.
- [91] S. Eftekharnejad, “Selection of multiple credible contingencies for real time contingency analysis,” in *2015 IEEE Power & Energy Society General Meeting*. IEEE, 2015, pp. 1–5.
- [92] I. H. Mads R. Almassalkhi. (2014) Model-predictive cascade mitigation in electric power systems with storage and renewables. [Online]. Available: https://www.ferc.gov/CalendarFiles/20140624104957-T3B%20-%201%20-%20Almassalkhi_Hiskens_Final.pdf
- [93] A.-A. Fouad and V. Vittal, *Power system transient stability analysis using the transient energy function method*. Pearson Education, 1991.
- [94] D. R. Gurusinge and A. D. Rajapakse, “Post-disturbance transient stability status prediction using synchrophasor measurements,” *IEEE Transactions on Power systems*, vol. 31, no. 5, pp. 3656–3664, 2015.
- [95] T. L. Vu and K. Turitsyn, “Lyapunov functions family approach to transient stability assessment,” *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1269–1277, 2015.

- [96] M. Pai and P. W. Sauer, "Stability analysis of power systems by Lyapunov's direct method," *IEEE Control Systems Magazine*, vol. 9, no. 1, pp. 23–27, 1989.
- [97] E. Farantatos, R. Huang, G. J. Cokkinides, and A. Meliopoulos, "A predictive generator out-of-step protection and transient stability monitoring scheme enabled by a distributed dynamic state estimator," *IEEE Transactions on Power Delivery*, vol. 31, no. 4, pp. 1826–1835, 2015.
- [98] Y. Xue, T. Van Cutsem, and M. Ribbens-Pavella, "Extended equal area criterion justifications, generalizations, applications," *IEEE Transactions on Power Systems*, vol. 4, no. 1, pp. 44–52, 1989.
- [99] M. Z. Jahromi and S. M. Kouhsari, "A novel recursive approach for real-time transient stability assessment based on corrected kinetic energy," *Applied Soft Computing*, vol. 48, pp. 660–671, 2016.
- [100] M. Pai, *Energy function analysis for power system stability*. Springer Science & Business Media, 2012.
- [101] P. Ju, H. Li, C. Gan, Y. Liu, Y. Yu, and Y. Liu, "Analytical assessment for transient stability under stochastic continuous disturbances," *IEEE Transactions on Power Systems*, vol. 33, no. 2, pp. 2004–2014, 2017.
- [102] J. Yan, C.-C. Liu, and U. Vaidya, "PMU-based monitoring of rotor angle dynamics," *IEEE Transactions on Power Systems*, vol. 26, no. 4, pp. 2125–2133, 2011.
- [103] M. Li, A. Pal, A. G. Phadke, and J. S. Thorp, "Transient stability prediction based on apparent impedance trajectory recorded by PMUs," *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 498–504, 2014.
- [104] R. Diao, V. Vittal, and N. Logic, "Design of a real-time security assessment tool for

- situational awareness enhancement in modern power systems,” *IEEE Transactions on Power systems*, vol. 25, no. 2, pp. 957–965, 2009.
- [105] T. Amraee and S. Ranjbar, “Transient instability prediction using decision tree technique,” *IEEE Transactions on power systems*, vol. 28, no. 3, pp. 3028–3037, 2013.
- [106] B. Wang, B. Fang, Y. Wang, H. Liu, and Y. Liu, “Power system transient stability assessment based on big data and the core vector machine,” *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2561–2570, 2016.
- [107] J. Geeganage, U. Annakkage, T. Weekes, and B. A. Archer, “Application of energy-based power system features for dynamic security assessment,” *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1957–1965, 2014.
- [108] Y. Xu, Z. Y. Dong, J. H. Zhao, P. Zhang, and K. P. Wong, “A reliable intelligent system for real-time dynamic security assessment of power systems,” *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1253–1263, 2012.
- [109] Y. Li and Z. Yang, “Application of EOS-ELM with binary Jaya-based feature selection to real-time transient stability assessment using PMU data,” *IEEE Access*, vol. 5, pp. 23 092–23 101, 2017.
- [110] S. K. Azman, Y. J. Isbeih, M. S. El Moursi, and K. Elbassioni, “A unified online deep learning prediction model for small signal and transient stability,” *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4585–4598, 2020.
- [111] L. Zhu, D. J. Hill, and C. Lu, “Hierarchical deep learning machine for power system online transient stability prediction,” *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2399–2411, 2019.
- [112] J. James, D. J. Hill, A. Y. Lam, J. Gu, and V. O. Li, “Intelligent time-adaptive tran-

- sient stability assessment system,” *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1049–1058, 2017.
- [113] Z. Li, H. Liu, J. Zhao, T. Bi, and Q. Yang, “Fast power system event identification using enhanced lstm network with renewable energy integration,” *IEEE Transactions on Power Systems*, 2021.
- [114] F. Pan, J. Li, B. Tan, C. Zeng, X. Jiang, L. Liu, and J. Yang, “Stacked-GRU based power system transient stability assessment method,” *Algorithms*, vol. 11, no. 8, p. 121, 2018.
- [115] Q. Zhu, J. Chen, L. Zhu, D. Shi, X. Bai, X. Duan, and Y. Liu, “A deep end-to-end model for transient stability assessment with PMU data,” *IEEE Access*, vol. 6, pp. 65 474–65 487, 2018.
- [116] D. You, K. Wang, L. Ye, J. Wu, and R. Huang, “Transient stability assessment of power system using support vector machine with generator combinatorial trajectories inputs,” *International Journal of Electrical Power & Energy Systems*, vol. 44, no. 1, pp. 318–325, 2013.
- [117] F. Da-Zhong, T. Chung, and A. David, “Fast transient stability estimation using a novel dynamic equivalent reduction technique,” *IEEE transactions on power systems*, vol. 9, no. 2, pp. 995–1001, 1994.
- [118] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient stability of power systems: a unified approach to assessment and control*. Springer Science & Business Media, 2012.
- [119] T. Guo and J. V. Milanović, “Probabilistic framework for assessing the accuracy of data mining tool for online prediction of transient stability,” *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 377–385, 2013.

- [120] G. Gong, N. K. Mahato, H. He, H. Wang, Y. Jin, and Y. Han, "Transient stability assessment of electric power system based on voltage phasor and cnn-lstm," in *2020 IEEE/IAS Industrial and Commercial Power System Asia (I&CPS Asia)*. IEEE, 2020, pp. 443–448.
- [121] A. D. Rajapakse, F. Gomez, K. Nanayakkara, P. A. Crossley, and V. V. Terzija, "Rotor angle instability prediction using post-disturbance voltage trajectories," *IEEE Transactions on Power Systems*, vol. 25, no. 2, pp. 947–956, 2009.
- [122] C. W. Taylor, D. C. Erickson, K. E. Martin, R. E. Wilson, and V. Venkatasubramanian, "Wacs-wide-area stability and voltage control system: R&d and online demonstration," *Proceedings of the IEEE*, vol. 93, no. 5, pp. 892–906, 2005.
- [123] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2019, pp. 5508–5518.
- [124] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [125] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: fine-grained image generation through asymmetric training," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2745–2754.
- [126] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.
- [127] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification," *Neurocomputing*, vol. 321, pp. 321–331, 2018.

- [128] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *arXiv preprint arXiv:1406.2661*, 2014.
- [129] C. Ren and Y. Xu, “A fully data-driven method based on generative adversarial networks for power system dynamic security assessment with missing data,” *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 5044–5052, 2019.
- [130] X. Zheng, B. Wang, and L. Xie, “Synthetic dynamic PMU data generation: A generative adversarial network approach,” in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, 2019, pp. 1–6.
- [131] X. Zheng, B. Wang, D. Kalathil, and L. Xie, “Generative adversarial networks-based synthetic pmu data creation for improved event classification,” *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 68–76, 2021.
- [132] R. Ma and S. Eftekharijad, “Data generation for rare transient events: A generative adversarial network approach,” *2021 IEEE Industry Applications Society Annual Meeting in proceedings*, 2021.
- [133] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *arXiv preprint arXiv:1409.2329*, 2014.
- [134] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International conference on machine learning*. PMLR, 2013, pp. 1310–1318.
- [135] L. K. Hansen and P. Salamon, “Neural network ensembles,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.
- [136] R. Ma, S. Basumallik, and S. Eftekharijad, “A PMU-based data-driven approach

for classifying power system events considering cyberattacks,” *IEEE Systems Journal*, vol. 14, no. 3, September 2020.

- [137] A. Al-Digs, S. V. Dhople, and Y. C. Chen, “Measurement-based sparsity-promoting optimal control of line flows,” *IEEE Trans. on Power Systems*, vol. 33, no. 5, pp. 5628–5638, 2018.
- [138] Y. Liu, N. Zhang, Y. Wang, J. Yang, and C. Kang, “Data-driven power flow linearization: A regression approach,” *IEEE Trans. on Smart Grid*, vol. 10, no. 3, pp. 2569–2580, 2018.
- [139] H. Noh, T. You, J. Mun, and B. Han, “Regularizing deep neural networks by noise: Its interpretation and optimization,” *arXiv preprint arXiv:1710.05179*, 2017.
- [140] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, “Learning from class-imbalanced data: Review of methods and applications,” *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.
- [141] B. Krawczyk, M. Woźniak, and G. Schaefer, “Cost-sensitive decision tree ensembles for effective imbalanced classification,” *Applied Soft Computing*, vol. 14, pp. 554–562, 2014.
- [142] S. Basumallik, R. Ma, and S. Eftekharnjad, “Packet-data anomaly detection in PMU-based state estimator using convolutional neural network,” *International Journal of Electrical Power & Energy Systems*, vol. 107, pp. 690–702, 2019.
- [143] M. A. Rahman and H. Mohsenian-Rad, “False data injection attacks with incomplete information against smart power grids,” in *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2012, pp. 3153–3158.
- [144] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estima-

- tion in electric power grids,” *ACM Trans. on Information and System Security*, vol. 14(1), pp. 1–33, 2011.
- [145] M. Esmalifalak, H. Nguyen, R. Zheng, and Z. Han, “Stealth false data injection using independent component analysis in smart grid,” in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2011, pp. 244–248.
- [146] A. Meliopoulos and A. Bakirtzis, “Corrective control computations for large power systems,” *IEEE transactions on power apparatus and systems*, no. 11, pp. 3598–3604, 1983.
- [147] L. Lenoir, I. Kamwa, and L.-A. Dessaint, “Overload alleviation with preventive-corrective static security using fuzzy logic,” *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 134–145, 2009.
- [148] A. Vergnol, J. Sprooten, B. Robyns, V. Rious, and J. Deuse, “Line overload alleviation through corrective control in presence of wind energy,” *Electric Power Systems Research*, vol. 81, no. 7, pp. 1583–1591, 2011.
- [149] W. Shao and V. Vittal, “Corrective switching algorithm for relieving overloads and voltage violations,” *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1877–1885, 2005.
- [150] A. A. Mazi, B. F. Wollenberg, and M. H. Hesse, “Corrective control of power system flows by line and bus-bar switching,” *IEEE Transactions on Power Systems*, vol. 1, no. 3, pp. 258–264, 1986.
- [151] P. Bijwe, D. Kothari, and L. Arya, “Alleviation of line overloads and voltage violations by corrective rescheduling,” in *IEE Proceedings C (Generation, Transmission and Distribution)*, vol. 140, no. 4. IET, 1993, pp. 249–255.

- [152] K. Bell, A. Daniels, and R. Dunn, "Modelling of operator heuristics in dispatch for security enhancement," *IEEE transactions on power systems*, vol. 14, no. 3, pp. 1107–1113, 1999.
- [153] M. Larsson, D. J. Hill, and G. Olsson, "Emergency voltage control using search and predictive control," *International Journal of Electrical Power & Energy Systems*, vol. 24, no. 2, pp. 121–130, 2002.
- [154] M. Zima and G. Andersson, "Model predictive control employing trajectory sensitivities for power systems applications," in *Proc. of the 44th IEEE Conference on Decision and Control*, 2005, pp. 4452–4456.
- [155] H. Jiang, J. Lin, Y. Song, and D. J. Hill, "Mpc-based frequency control with demand-side participation: A case study in an isolated wind-aluminum power system," *IEEE Transactions on Power Systems*, vol. 30, no. 6, pp. 3327–3337, 2014.
- [156] T. H. Mohamed, J. Morel, H. Bevrani, and T. Hiyama, "Model predictive based load frequency control design concerning wind turbines," *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 859–867, 2012.
- [157] B. Otomega, A. Marinakis, M. Glavic, and T. Van Cutsem, "Model predictive control to alleviate thermal overloads," *IEEE Trans. on Power Systems*, vol. 22, no. 3, pp. 1384–1385, 2007.
- [158] B. Otomega, A. Marinakis, M. Glavic, and T. Van Cutsem, "Emergency alleviation of thermal overloads using model predictive control," in *2007 IEEE Lausanne Power Tech.* IEEE, 2007, pp. 201–206.
- [159] J. S. Carneiro and L. Ferrarini, "Preventing thermal overloads in transmission circuits via model predictive control," *IEEE Trans. on Control Systems Technology*, vol. 18, no. 6, pp. 1406–1412, 2010.

- [160] M. R. Almassalkhi and I. A. Hiskens, “Model-predictive cascade mitigation in electric power systems with storage and renewables—part i: Theory and implementation,” *IEEE Trans. on Power Sys.*, vol. 30(1), pp. 67–77, 2014.
- [161] M. R. Almassalkhi and I. A. Hiskens, “Model-predictive cascade mitigation in electric power systems with storage and renewables—part ii: Case-study,” *IEEE Trans. on Power Systems*, vol. 30(1), pp. 78–87, 2014.
- [162] J. A. Martin and I. A. Hiskens, “Corrective model-predictive control in large electric power systems,” *IEEE Trans. on Power Systems*, vol. 32, no. 2, pp. 1651–1662, 2016.
- [163] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [164] K. Van den Bergh, E. Delarue, and W. D’haeseleer, “Dc power flow in unit commitment models,” *no. May*, 2014.
- [165] Y. C. Chen, A. D. Domínguez-García, and P. W. Sauer, “Measurement-based estimation of linear sensitivity distribution factors and applications,” *IEEE Trans. on Power Sys.*, vol. 29(3), pp. 1372–1382, 2014.
- [166] R. Isermann and M. Münchhof, *Identification of dynamic systems*. Springer Science & Business Media, 2010.
- [167] K. J. Keesman, *System identification: an introduction*. Springer Science & Business Media, 2011.
- [168] J. C. Willems, P. Rapisarda, I. Markovskiy, and B. L. De Moor, “A note on persistency of excitation,” *Systems & Control Letters*, vol. 54, no. 4, pp. 325–329, 2005.

- [169] I. Markovsky and P. Rapisarda, “Data-driven simulation and control,” *International Journal of Control*, vol. 81, no. 12, pp. 1946–1959, 2008.
- [170] F. Kong, M. Xu, J. Weimer, O. Sokolsky, and I. Lee, “Cyber-physical system checkpointing and recovery,” in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems*, pp. 22–31.
- [171] R. Ma, S. B. Mallik, and S. Eftekharnjad, “A pmu-based multivariate model for classifying power system events,” *IEEE Systems Journal*, 12 2018.
- [172] G. Hug and J. A. Giampapa, “Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks,” *IEEE Trans. on smart grid*, vol. 3, no. 3, pp. 1362–1370, 2012.
- [173] G. N. Korres and N. M. Manousakis, “State estimation and bad data processing for systems including PMU and SCADA measurements,” *Electric Power Systems Research*, vol. 81, no. 7, pp. 1514–1524, 2011.
- [174] M. Ciocca, P.-B. Wieber, and T. Fraichard, “Strong recursive feasibility in model predictive control of biped walking,” in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics*, 2017, pp. 730–735.
- [175] R. Ma, S. Basumallik, S. Eftekharnjad, and F. Kong, “Recovery-based model predictive control for cascade mitigation under cyber-physical attacks,” *IEEE Texas Power and Energy Conference*, 2020.
- [176] Y. Chen, A. Domínguez-García, and P. W. Sauer, “A sparse representation approach to online estimation of power system distribution factors,” *IEEE Trans. on Power Sys.*, vol. 30(4), pp. 1727–1738, 2014.
- [177] 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 Trans. on Power Sys.*, vol. 26, no. 1, pp. 12–19, 2011.

- [178] Y. Chen, A. Domínguez-García, and P. W. Sauer, “A sparse representation approach to online estimation of power system distribution factors,” *IEEE Trans. on Power Sys.*, vol. 30 (4), pp. 1727–1738, 2015.
- [179] “Active power ramp rates,” https://neos-guide.org/sites/default/files/ramp_rates.pdf, accessed: 2020-11-25.
- [180] J. Berberich and F. Allgöwer, “A trajectory-based framework for data-driven system analysis and control,” *arXiv:1903.10723*, 2019.
- [181] NERC. (2014) System operating limit definition and exceedance clarification. [Online]. Available: <https://bit.ly/2J7fcbq>
- [182] ISU. The dc power flow equations. [Online]. Available: <http://home.eng.iastate.edu/~jdm/ee553/DCPowerFlowEquations.pdf>
- [183] S. Wang, J. Zhao, Z. Huang, and R. Diao, “Assessing gaussian assumption of PMU measurement error using field data,” *IEEE Trans. on Power Delivery*, vol. 33, no. 6, pp. 3233–3236, 2017.
- [184] L. Ding, P. Hu, Z. Liu, and G. Wen, “Transmission lines overload alleviation: Distributed online optimization approach,” *IEEE Trans. on Industrial Informatics*, 2020.

Vita

Name of Author: Rui Ma

Place of Birth: Shanxi, China

Major: Electrical & Computer Engineering

Education

- Ph.D., Syracuse University, Syracuse, USA, 2022
- M.S., The Ohio State University, Columbus, USA, 2015
- B.S., Central South University, Changsha, China, 2014

Professional Experience

- Research Intern, Global Energy Interconnection Research Institute North America, San Jose, USA, May 2020 - August 2020
- Graduate Assistant, Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, USA, August 2016- May 2020
- Teaching Assistant, Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, USA, August 2020- May 2022

Awards and Honors

- Third Prize of Graduate Student Research Competition, 2021

- Syracuse GSO Travel Grant, 2020 and 2021
- Student Registration Grant Award in IAS Annual Meeting, 2021
- IEEE Student Grant, 2018 IEEE/PES Transmission and Distribution Conference and Exposition (TD)
- IEEE Student Grant, 2019 IEEE Power Energy Society General Meeting (PESGM)