A Data Analytics Framework for Smart Grids: Spatio-temporal Wind Power Analysis and Synchrophasor Data Mining

by

Miao He

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

Approved July 2013 by the
Graduate Supervisory Committee:

Junshan Zhang, Co-Chair
Vijay Vittal, Co-Chair
Kory Hedman
Jennie Si
Jieping Ye

ARIZONA STATE UNIVERSITY

August 2013

ABSTRACT

Under the framework of intelligent management of power grids by leveraging advanced information, communication and control technologies, a primary objective of this study is to develop novel data mining and data processing schemes for several critical applications that can enhance the reliability of power systems. Specifically, this study is broadly organized into the following two parts: I) spatio-temporal wind power analysis for wind generation forecast and integration, and II) data mining and information fusion of synchrophasor measurements toward secure power grids.

Part I is centered around wind power generation forecast and integration. First, a spatio-temporal analysis approach for short-term wind farm generation forecasting is proposed. Specifically, using extensive measurement data from an actual wind farm, the probability distribution and the level crossing rate of wind farm generation are characterized using tools from graphical learning and time-series analysis. Built on these spatial and temporal characterizations, finite state Markov chain models are developed, and a point forecast of wind farm generation is derived using the Markov chains. Then, multi-timescale scheduling and dispatch with stochastic wind generation and opportunistic demand response is investigated.

Part II focuses on incorporating the emerging synchrophasor technology into the security assessment and the post-disturbance fault diagnosis of power systems. First, a data-mining framework is developed for on-line dynamic security assessment by using adaptive ensemble decision tree learning of real-time synchrophasor measurements. Under this framework, novel on-line dynamic security assessment schemes are devised, aiming to handle various factors (including variations of operating conditions, forced system topology change, and loss of critical synchrophasor measurements) that can have significant impact on the performance of conventional data-mining based on-

line DSA schemes. Then, in the context of post-disturbance analysis, fault detection and localization of line outage is investigated using a dependency graph approach. It is shown that a dependency graph for voltage phase angles can be built according to the interconnection structure of power system, and line outage events can be detected and localized through networked data fusion of the synchrophasor measurements collected from multiple locations of power grids. Along a more practical avenue, a decentralized networked data fusion scheme is proposed for efficient fault detection and localization.

To my dear family.

ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my advisors, Professor Junshan Zhang and Professor Vijay Vittal, who have continually guided and supported me, and furthered my professional and personal development throughout my stay at Arizona State University. They are wonderful mentors, from whom I learnt the methodologies of conducting research and many other things. Their constant passion and distinctive visions for research will be an everlasting inspiration for me to pursue my own potentials. It is my greatest fortune to have them as Ph.D. advisors.

I would also like to give my sincere appreciation to Professors Kory Hedman, Jennie Si and Jieping Ye, for their valuable time in serving on my committee and all the insightful comments and suggestions they have made. Special thanks to my collaborators, Dr. Lei Yang and Dr. Sugumar Murugesan, for their inspiring discussions and generous help. Many thanks to my colleagues in the Information and Energy Networks group: Weiyan Ge, Qinghai Gao, Ashwin Kowdle, Huilan Zou, Jia Lou, Dong Zheng, Tuan Tran, Chandrashekar Theajaswi P.S., Osman Yagan, Kai Cai, Shanshan Wang, Ling Tang, Dajun Qian, Brian Proulx, Xiaowen Gong, Yang Cao, David Ganger, Mojgan Hedayati, Shibo He, Chuan Huang, Donghoon Shin, Xu Chen, Zhengyu Zhang, and Mengyuan Zhang for the pleasant time they spent together with me.

Finally, I dedicate this dissertation to my dear family, for their endless love and support. It is such a blessing to have them on my side in the unforgettable journey of pursuing my Ph.D. degree!

TABLE OF CONTENTS

Page

LIST OF FIGURES xiv

LIST OF TABLES xviii

NOMENCLATURE xix

CHAPTER

1 INTRODUCTION 1

1.1 Big Data in Smart Grid 1

1.2 Synchrophasor Technology 3

1.3 Reliability of Power Systems 5

1.3.1 Impact of Wind Generation Integration on Power System . . . 6

1.4 Scope of The Dissertation and Summary of Main Contributions . . . 6

1.4.1 Part I: Spatio-temporal Analysis for Wind Farm Generation Forecast and Integration 6

1.4.2 Part II: Synchrophasor Data Mining and Information Fusion . 8

PART I SPATIO-TEMPORAL ANALYSIS FOR WIND FARM GENERATION FORECAST AND INTEGRATION 10

2 SPATIO-TEMPORAL ANALYSIS FOR SHORT-TERM WIND FARM GENERATION FORECAST 12

2.1 Introduction 12

2.1.1 Summary of Main Results and Contributions 13

CHAPTER Page

2.2 Available Data and Key Observations 15

2.2.1 Spatial Dynamics of Wind Farm 16

2.2.2 Diurnal Non-Stationarity and Seasonality 17

2.2.3 Weibull Distribution of Wind Speed 19

2.3 Spatio-temporal Analysis of Wind Farm Generation 20

2.3.1 Spatial Analysis and Statistical Characterization 20

2.3.2 Temporal Analysis and LCR Quantification 23

2.3.3 Markov Chain Model for Spatio-temporal Wind Power 24

2.4 Markov-Chain-based Short-term Forecast of Wind Farm Generation . 28

2.4.0.1 Point Forecasts 29

2.4.1 Numerical Examples 30

2.4.1.1 Distributional Forecasts 30

2.4.1.2 Point Forecasts 34

2.5 Conclusion 38

3 MULTI-TIMESCALE SCHEDULING WITH STOCHASTIC WIND GENERATION AND OPPORTUNISTIC DEMAND 39

3.1 Introduction 39

3.2 A Multi-timescale Scheduling Framework 41

3.2.1 General Problem Formulation 42

CHAPTER Page

3.2.2 Energy Supply and Demand Models 43

3.2.2.1 Stochastic wind generation 43

3.2.2.2 Generation cost 44

3.2.2.3 Uncertain demand of traditional energy users 44

3.2.2.4 Uncertain demand of opportunistic energy users . . . 45

3.2.2.5 Day-Ahead and Real-Time Pricing 45

3.2.3 Net Profit in a T_2-slot 46

3.3 Multi-timescale Scheduling with Non-persistent Opportunistic Energy Users 47

3.3.1 Real-time Scheduling on Timescale T_2 47

3.3.2 Day-ahead Scheduling on Timescale T_1 48

3.3.3 Approximate Solutions 48

3.3.3.1 Approximate Solution to the Real-time Scheduling Problem 50

3.3.3.2 Approximate Solution to the Day-ahead Scheduling Problem 52

3.4 Multi-timescale Scheduling with Persistent Opportunistic Energy Users 53

3.5 Numerical Results 57

3.5.1 Simulation Data 58

3.5.1.1 Stochastic Wind Generation 58

CHAPTER Page

3.5.1.2 Generation Cost 58

3.5.1.3 Uncertain Demand 59

3.5.2 Performance Evaluation and Discussion 59

3.5.2.1 Per Unit Generation Cost 60

3.5.2.2 Reliability 61

3.5.2.3 Impact of Forecast Accuracy 62

3.6 Conclusion 63

PART II SYNCHROPHASOR DATA MINING AND INFORMATION FUSION 63

4 A ROBUST DATA-MINING FRAMEWORK FOR ON-LINE DSA USING ADAPTIVE ENSEMBLE DECISION TREE LEARNING 65

4.1 Introduction 65

4.2 Background on DT and Its Application to DSA 68

4.2.1 Small DTs 70

4.2.2 Ensemble of DTs 71

4.2.3 Updating DTs 72

4.3 Proposed Scheme for Online DSA 72

4.3.1 Offline Training 73

4.3.1.1 Initial Knowledge Base Preparation 73

4.3.1.2 Boosting Small DTs 75

CHAPTER Page

4.3.2 Periodic Updates 77

4.3.2.1 New Training Case Creation 77

4.3.2.2 Updating the Classification Model 77

4.3.3 Online DSA 78

4.4 An Illustrative Example 80

4.4.1 Knowledge Base 81

4.4.1.1 Attribute Selection 81

4.4.1.2 OC Generation and Contingencies 81

4.4.1.3 Transient Stability Assessment Tool and Criteria . . 82

4.4.2 Offline Training 83

4.4.2.1 Choice of J and L 83

4.4.2.2 Ensemble Small DT Learning 83

4.4.3 Robustness Testing 84

4.4.3.1 Changed OCs 84

4.4.3.2 Robustness Testing Results 85

4.5 Application to the WECC System 86

4.5.1 Knowledge Base 86

4.5.1.1 OC Generation 86

4.5.1.2 Critical Contingency Selection 88

CHAPTER Page

4.5.1.3 Case Creation 88

4.5.2 Offline Training 89

4.5.3 Online DSA Simulation 89

4.5.3.1 OC variations in sub-period 12:00 Hrs-16:00 Hrs . . . 90

4.5.3.2 Topology change in sub-period 16:00 Hrs-20:00 Hrs . 90

4.5.4 Test Results and Discussion 90

4.5.4.1 Classification Accuracy 91

4.5.4.2 Computation Requirement 91

4.6 Conclusion 92

5 ROBUST ON-LINE DSA WITH MISSING SYNCHROPHASOR DATA . 93

5.1 Introduction 93

5.2 Background on DTs 96

5.2.1 Application of DTs in DSA 97

5.2.2 Handling Missing Data by using Surrogate in DTs 97

5.2.3 Ensemble DT Learning 99

5.3 Random Subspace Method for Selecting Attribute Subsets 100

5.3.1 Candidate Attribute Subsets 101

5.3.2 Randomized Algorithm for Selecting Attribute Subsets 102

5.4 Proposed Approach for Online DSA with Missing PMU Measurements 105

CHAPTER Page

5.4.1 Offline Training 105

5.4.2 Near Real-time Performance Re-check 106

5.4.3 Online DSA 106

5.4.3.1 Degenerate Small DTs 107

5.4.3.2 Weighted Voting of Viable Small DTs 107

5.4.4 Further Discussion 109

5.5 Case Study 110

5.5.1 Test System 110

5.5.2 Knowledge base 111

5.5.2.1 OC Generation 112

5.5.2.2 Contingencies 112

5.5.2.3 Transient Security Assessment 113

5.5.3 Performance Evaluation 113

5.5.3.1 Attribute Subsets 114

5.5.3.2 Offline Training 114

5.5.3.3 Near Real-time Re-check 115

5.5.3.4 Online DSA Test 115

5.5.3.5 Impact of Measurement Noise 117

5.6 Conclusion 118

CHAPTER Page

6 A DEPENDENCY GRAPH APPROACH FOR FAULT DETECTION AND LOCALIZATION USING SYNCHROPHASOR DATA 119

6.1 Introduction 119

6.2 Dependency Graph Models for PMU Data 121

6.2.1 Background on Dependency Graph 121

6.2.2 Gaussian Markov Random Field for Phase Angles 123

6.3 Decentralized Network Inference for Fault Detection and Localization 125

6.3.1 Two-scale Decomposition of GMRF 129

6.3.2 Multiscale Decomposition of GMRF 132

6.3.3 Decentralized Network Inference using Message Passing 133

6.4 Numerical Results 135

6.4.1 Decentralized Estimation of Information Matrix 135

6.4.2 Fault Localization 136

6.4.2.1 Decomposition of the IEEE 300-bus System 138

6.4.2.2 Testing cases for line outage 138

6.4.2.3 Testing cases for the change of physical parameter . 139

6.5 Conclusion 139

7 Future Research 141

7.1 Impact of Communication QoS on the Performance of Special Protection Schemes 141

CHAPTER Page

7.2 Near Real-time Prediction and Quantification of Extreme Ramp-down Events of Wind Farm Generation 142

7.3 Data Security and Privacy in Smart Grids 143

REFERENCES 145

APPENDIX 157

A DERIVATIONS FOR CHAPTER 3 157

A.1 Proof of Proposition 3.3.1 158

A.2 Proof of Proposition 3.3.2 158

A.3 Proof of Proposition 3.3.3 158

B DERIVATIONS FOR CHAPTER 4 160

B.1 Derivation for $\hat{\mathrm{e}}(H_L) \leq C_N(F_L)$ 161

B.2 Derivation for Convexity of $C_N(F_L)$ 161

B.3 Derivation for Differentiability of $C_N(F_L)$ 162

B.4 Derivation for Convergence of F_L 163

B.5 Derivation for Uniqueness of the minimum of $G_N^{(l)}(a)$ 163

C DERIVATIONS FOR CHAPTER 6 164

C.1 Proof of Lemma 6.3.1 165

C.2 Proof of Lemma 6.3.2 167

C.3 Proof of Proposition 6.3.1 168

LIST OF FIGURES

Figure Page

1.1 Big data emerging in smart grids [1]. 2

1.2 Phasor representation of bus voltage with a common reference [2]. 3

1.3 PMU block diagram [2]. 4

2.1 Power curves for wind turbines from classes C_1 and C_2. 15

2.2 Power outputs of three wind turbines in C_1. 16

2.3 Tail probability of the wind farm's aggregate power output. 17

2.4 Empirical distribution of wind farm generation over various 1-hour intervals of different epochs of the day and different months. 18

2.5 Weibull-fitted CDF (λ=11.37, k=1.54) and empirical CDF of W_1 for the 9 AM-noon epoch of January 2009. 19

2.6 MST of C_1 (with distance to the southwest corner of the wind farm). . . 20

2.7 Power outputs of parent-child turbine pairs of C_1 for the 9 AM-noon epoch of January 2009. 21

2.8 CDF of $P_{ag}(t)$ for the 9 AM-noon epoch of January 2009. 22

2.9 LCR of $P_{ag}(t)$ for the 9 AM-noon epoch of January 2009. 25

2.10 boundaries and average duration for each state of the Markov chain for the 9 AM-noon epoch of January 2009. 27

2.11 Transition matrix (a) by spatio-temporal analysis (b) by uniform quantization, for the 9 AM-noon epoch of January 2009. 28

Figure Page

2.12 Offline spatio-temporal analysis (carried out for each epoch and each month by using historic measurement data). 28

2.13 Online short-term forecasting. 29

2.14 Statistics of CRPS over all months of the year 2010. 31

2.15 Statistics of CRPS over all 8 epochs of the year 2010. 32

2.16 Correlation between the ramp rates of $P_{ag}(t)$ and the CRPS values of distributional forecast. 33

2.17 10-min distributional forecasts on January 8th, 2010. 34

2.18 10-min distributional forecasts on January 23rd, 2010. 35

2.19 10-min distributional forecasts on April 16th, 2010. 35

2.20 Number of states and the forecast error of Markov chains at various τ for the January 9 AM-noon epoch). 37

3.1 A multi-timescale scheduling framework with integrated wind generation and two classes of energy users 41

3.2 PDF of wind generation forecast error in the mth T_1-slot. 59

3.3 Per unit generation cost ($\lambda = 20\%$). 60

3.4 Dispatched fast-start generation ($\lambda = 20\%$). 60

3.5 System Reliability ($\lambda = 20\%$). 61

3.6 Non-spinning reserve requirement ($\eta_w = 20\%$). 62

Figure Page

4.1 A fully-grown DT of height 5 for the WECC system using an initial knowledge base consisting of 481 OCs and three critical contingencies. 68

4.2 The first three small DTs (J=2) for the WECC system, the voting weights of which are 4.38, 3.04 and 0.93, respectively. 70

4.3 Proposed scheme for online DSA using adaptive ensemble DT learning. . 73

4.4 Boosting small DTs 75

4.5 The IEEE 39-bus system with 8 PMUs 80

4.6 Ensemble small DT learning with different tree heights for the IEEE 39-bus test system. 82

4.7 The first small DT h_1 (J=2) for the IEEE 39-bus test system. 84

4.8 Aggregate load of recorded OCs and generated OCs by interpolation. . . 87

4.9 Flowchart for testing online DSA with periodic updating. 89

4.10 Computation time for updating/rebuilding (executed in MATLAB on a workstation with an Intel Pentium IV 3.20 GHz CPU and 4GB RAM). . 91

5.1 Three-stage ensemble DT-based approach to online DSA with missing PMU measurements 95

5.2 Wide area monitoring system consisting of multiple areas 100

5.3 Degeneration of a small DT as a result of missing PMU measurements of attribute x_1 when node (x_1<S_1) is originally assigned +1. 107

5.4 The IEEE 39-bus system in three areas and PMU placement 111

5.5 Performance of online DSA in case of missing PMU measurements 116

Figure Page

5.6 Impact of measurement noise. 118

6.1 The IEEE 14-bus system and the dependency graph of the phase angles . 123

6.2 The two-scale decomposition of GMRF $\mathbf{X}$ 129

6.3 The multiscale decomposition of GMRF $\mathbf{X}$ 132

6.4 The extended sub-field for the fault diagnosis of S_3 135

6.5 The performance of estimation algorithms ($\gamma = 0.5$) 136

6.6 The performance of estimation algorithms ($d_0 = 4$) 137

6.7 The top-scale sub-field of the IEEE 300-bus System 138

LIST OF TABLES

Table Page

2.1 Skill values of Markov-chain-based distributional forecasts over the testing data points of the year 2010. 31

2.2 10-min point forecast error of wind farm generation over the period shown in Fig.2.17. 36

2.3 10-min point forecast error of wind farm generation over the period shown in Fig.2.18. 36

2.4 10-min point forecast error of wind farm generation over the period shown in Fig.2.19. 37

2.5 10-min point forecast error of wind farm generation (all test data of the year 2010 is used). 37

4.1 Misclassification Error Rate of Robustness Testing 86

4.2 Misclassification Error Rate of Online DSA 92

5.1 A case study on the surrogates of DTs 99

5.2 Data used by Algorithm 1 for the system in Fig. 5.4. 113

5.3 The first and second removed components of the selected $N-2$ contingencies 115

5.4 Misclassification error rate when b=0.94. 116

6.1 A case of line {26,27} outage 139

6.2 A case of parameter changes at line {26,27} and {199,200} 139

NOMENCLATURE

AMI	advanced metering infrastructure
AR	auto-regressive model
CAR	conditional auto-regression model
CART	classification and regression tree
CDF	cumulative distribution function
CLT	central limit theorem
CSR	critical splitting rule
DER	distributed energy resource
DSA	dynamic security assessment
DSM	demand side management
DT	decision tree
EENS	expected energy not served
GMRF	Gaussian Markov random field
ICT	information and communications technology
IED	intelligent electronic device
LAN	local area network
LCR	level crossing rate
LOLP	loss of load probability
LPW	linear prediction of windspeed
MAE	mean absolute error
MAPE	mean absolute percentage error
MDP	Markov decision process
MET	meteorological tower
MMDP	multi-timescale Markov decision process

MMSE	minimum mean square error
MRF	Markov random field
NERC	North American Electric Reliability Corporation
NIST	National Institute of Standards and Technology
OC	operating condition
PDC	phasor data concentrator
PDF	probability distribution function
PMU	phasor measurement unit
PPW	persistent prediction of windspeed
PSAT	power flow and short circuit analysis tool
RAS	remedial action scheme
RF	random forest
RMS	root mean square
RMSE	root mean square error
RPS	renewable portfolio standard
RTU	remote terminal unit
SCADA	supervisory control and data acquisition
SPS	special protection scheme
TSAT	transient security assessment tool
TSI	transient stability index
TVE	total vector error
WAMPAC	wide-area monitoring, protection and control
WAMS	wide area monitoring system
WAN	wide-area network
WECC	Western Electricity Coordinating Council

Chapter 1

INTRODUCTION

The age of a "smart" grid is upcoming. In the past decades, tremendous national efforts have been dedicated to building the elements of a modern electric power grid, which will ultimately become a smart grid. Futuristic smart grids are envisaged to dramatically increase the efficiency of electricity production and distribution, reduce greenhouse gas emissions and support a sustainable energy infrastructure. According to the Energy Independence and Security Act [3], some key distinguishing characteristics of smart grids are given below:

- increased use of digital information and communication technology (ICT) concerning grid status and situational awareness;
- deployment of advanced wide-area monitoring, protection and control (WAMPAC) schemes to improve the reliability and the efficiency of electric power grids;
- high penetration of renewable power generation and distributed energy resources (DER).

The aforementioned three characteristics of smart grids define the scope of this study. In what follows, some basic concepts are first introduced.

1.1 Big Data in Smart Grid

The future smart grid is expected to leverage advanced ICTs to improve grid operations and planning. For example, widely dispersed phasor measurement units (PMUs) and PMU-enabled intelligent electronic devices (IEDs) make WAMPAC possible; smart meters and advanced metering infrastructure (AMI) enable two-way communications between end-users and utilities and sophisticated demand side manage-

ment; at wind farms, measurement data of individual turbine's power output at high temporal resolutions facilitates wind power generation analysis and forecast.

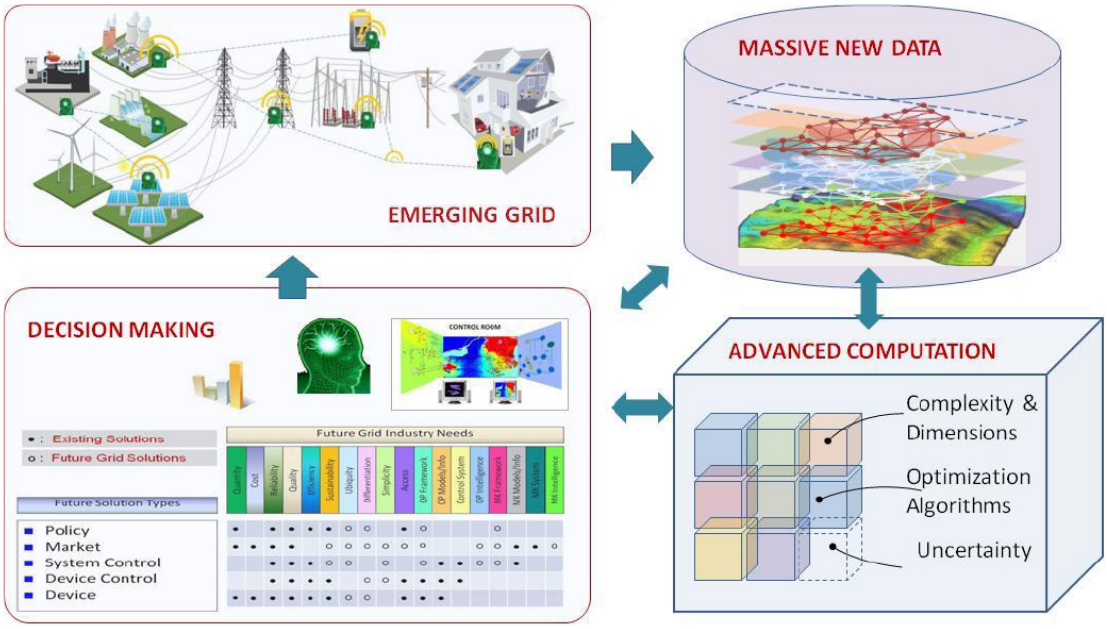


Figure 1.1: Big data emerging in smart grids [1].

Therefore, future smart grids have the potential to generate massive amounts of data from widely deployed measurement devices. Massive amounts of more detailed data collected from networked measurement devices provides great opportunities for enhanced situational awareness. On the other hand, it also raises new challenges for the effective extraction of relevant information from massive data so as to support decision making. Under increasingly dynamic and uncertain conditions of the power grid, new computational methods are necessary for efficient management of massive data. New algorithms for data fusion, data mining and data analytics have to be developed based on deeper understanding of the spatial and temporal dynamics of power systems.

1.2 Synchrophasor Technology

A phasor is a representation of sinusoidal signals using magnitude and phase. In power system monitoring and control applications, bus voltage and branch current are usually represented as phasors. Given the sinusoidal function of a voltage/current signal, the magnitude of the voltage/current phasor is the effective value, i.e., the root mean square (RMS) value of the sinusoidal function, and the phase of the voltage/current phasor is calculated with regard to a specified reference, by using an angular measure. Fig. 1.2 illustrates the phasor representations of the voltages measured at two terminal buses of a transmission line. Here, the common reference for the phase angles is 90 degrees (at time 0).

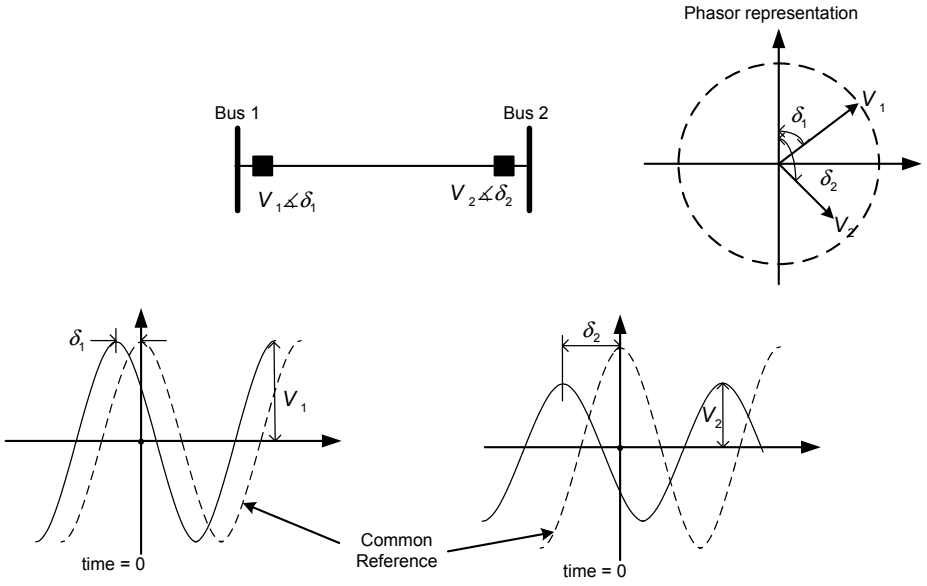


Figure 1.2: Phasor representation of bus voltage with a common reference [2].

The phasor measurements collected from widely dispersed locations of a power grid are precisely synchronized to a common reference, by using GPS receiver embedded in PMUs. The block diagram of PMUs is provided in Fig. 1.3. Basically, a PMU is an electronic device that uses state-of-the-art digital signal processing techniques

that converts analog waveforms from current/potential transformers into digital data. This analog-to-digital conversion is typically carried out at a rate of 48 samples per cycle (2880 samples per second), and the output rate of PMU data can reach up to 60 samples per second. A phase-locked oscillator, together with a GPS reference source with 1 microsecond accuracy, guarantees that all phasor measurements are time-tagged and all phase angles are synchronized to the same reference. By use of synchronized phasor (thus called "synchrophasor") data, the system-wide snapshots of a power grid can be obtained at a very high temporal resolution.

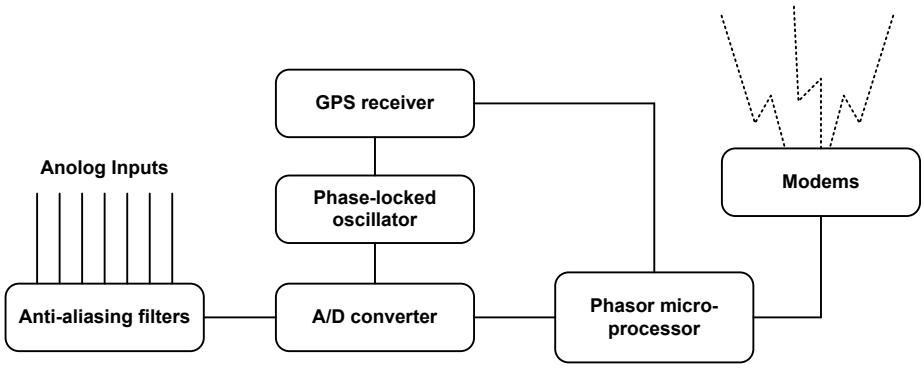


Figure 1.3: PMU block diagram [2].

Synchrophasor technology has been recognized as one of the most critical measurement technologies of smart grids. Compared to existing supervisory control and data acquisition (SCADA) systems, wide area monitoring systems (WAMS) built on synchrophasor technology offer greater flexibility to WAMPAC applications:

- The precise synchronization of phasor measurement data offers wide-area visibility of power grids. Further, it also makes distributed synchrophasor data processing and information fusion possible, facilitating coordinated decision making and control actions.
- In conventional SCADA systems, grid status awareness is achieved through static state analysis, and therefore the dynamic behavior of power grids cannot

be observed. Whereas in WAMS, synchrophasor data is typically collected at 20, 30 or 60 samples per second. Therefore, the limitations of conventional SCADA systems are overcome by the high sub-second visibility of power grids provided by synchrophasor data.

- PMUs directly measure the phasor angles of bus voltage and branch current. In conventional SCADA systems, those values are obtained from static state estimation, which is highly susceptible to unsynchronized measurements and bad data.

- Synchrophasor data provides enhanced post-contingency assessment capability and can improve the agility of protective and corrective control actions.

1.3 Reliability of Power Systems

For bulk power systems, reliability is defined by the North American Electric Reliability Corporation (NERC) as the capability to meet end-users' electricity demand with a reasonable assurance of continuity and quality [4]. Further, NERC subdivides the reliability of power system into adequacy and security. Specifically, power system adequacy refers to "having sufficient resources to provide customers with a continuous supply of electricity at the proper voltage and frequency, virtually all of the time" [4]. Here, the "resources" include a combination of generation, transmission and distribution facilities, as well as demand side management (DSM) which may reduce end-users' electricity demand as needed. Power system security relates to the ability of bulk power systems to withstand unexpected disturbances, such as transmission line outage, loss of generator and transformer failure.

1.3.1 Impact of Wind Generation Integration on Power System

During the last decade, wind power has been the fastest in growth among all renewable energy resources [5]. With a prospective high penetration level, wind generation integration is expected to change dramatically the existing operating practices (e.g., unit commitment, economic dispatch and ancillary services procurement) that are critical to the adequacy of bulk power systems. Compared to conventional generation (e.g., thermal, hydro, nuclear), wind generation has three distinct characteristics: non-dispatchability, variability and uncertainty. Wind generation is generally non-dispatchable, in the sense that the power output of a wind farm cannot be simply dispatched at the request of power grid operators. This is because, unlike conventional generators, the "fuel" of wind turbines, i.e., wind, cannot be controlled or stored. Due to the aforementioned characteristics, timely and accurate wind generation forecasting is critical to ensure that adequate resources for dispatch, ancillary services and ramping requirements are available all the time. Several balancing authorities in North America have been implementing wind generation forecasting systems [6].

1.4 Scope of The Dissertation and Summary of Main Contributions

1.4.1 Part I: Spatio-temporal Analysis for Wind Farm Generation Forecast and Integration

In Chapter 2, a Markovian model for wind farm generation forecast is developed. This model takes into account both the spatial and temporal dynamics of the power outputs from widely dispersed turbines within a large wind farm. Using extensive data from a wind farm in the western U.S., a spatio-temporal analysis of the aggregate wind generation from the farm is performed. It is observed from actual measurement data that the power outputs from the turbines are often not equal. Motivated by

this observation and using tools from graphical learning, a rigorous step-by-step procedure to characterize the probability distribution of the aggregate wind generation from the farm is carried out, while the diurnal non-stationarity and the seasonality of wind speed are also accounted for. The temporal dynamics of the aggregate wind generation are then characterized using auto-regression analysis. Built on these spatial and temporal characterizations, a finite state Markov chain for aggregate wind generation forecast is then derived in a rigorous optimization framework. Numerical study using realistic data demonstrates the significant gain in the forecast accuracy by using the designed Markov chain compared to the wind-speed-based forecasting methods. The procedure developed here is amenable to the case when the farm has turbines from multiple classes, e.g., when they belong to multiple manufacturers or when they are deployed at different hub heights.

Chapter 3 addresses the challenges of integrating volatile wind generation into the bulk power grid. Notably, wind generation is among the renewable resources that has most variability and uncertainty, and exhibits multi-level dynamics across time. To enhance the penetration of wind energy, multi-timescale scheduling and dispatch of both conventional energy sources (e.g., thermal) and wind generation is studied, for a smart grid model with two classes of energy users, namely traditional energy users and opportunistic energy users (e.g., smart meters or smart appliances). Specifically, the system operator performs scheduling at two timescales. In day-ahead scheduling, with the statistical information on wind generation and energy demands, the operator optimally procures conventional energy supply and decides the optimal retail price for the traditional energy users for the next day. In real-time scheduling, upon the realization of the wind energy generation and the demand from traditional energy users (which is stochastically dependent on the day-ahead retail price), the operator decides the real-time retail price for the opportunistic energy users. In particular,

two types of opportunistic energy users are considered: non-persistent and persistent users. The non-persistent users leave the power market when they find that the current real-time price is unacceptable, whereas the persistent opportunistic users wait for the next acceptable real-time price. Closed-form solutions for the scheduling problem are obtained for non-persistent case. For the persistent case, the scheduling problem is first formulated as a multi-timescale Markov decision process (MMDP), which is then recast explicitly as a standard Markov decision process (MDP) that can be solved via standard solution techniques. Numerical results demonstrate that the proposed two-timescale scheduling and dispatch improves the overall efficiency with high penetration of wind generation, by enabling two-way energy exchange between energy providers and end-users, and by facilitating both information interaction as well as energy interaction.

1.4.2 Part II: Synchrophasor Data Mining and Information Fusion

In Chapter 4, a robust on-line dynamic security assessment (DSA) scheme is developed by using adaptive ensemble decision tree (DT) learning, with an objective to handle the variations of operating condition (OC) and system topology changes. Specifically, the classification model for DSA is built on boosting multiple unpruned small-height DTs. In off-line training, the small DTs and their voting weights are sequentially identified in a "gradient-descent" manner to minimize the misclassification cost. The small DTs, together with their voting weights, are then periodically updated throughout the operating horizon, by using new training cases that are created to account for any change in OC or network topology. Different from conventional DT-based DSA schemes, the training cases are assigned different data weights by each small DT; and higher data weights are assigned to a new training case if it is misclassified by the small DTs. The aforementioned techniques are utilized to minimize the misclassification cost as new training cases are added to the knowledge base, so that

the classification model can smoothly track the changes in OCs or system topology. The results from a case study using a variety of realistic OCs illustrate the effectiveness of the proposed scheme in dealing with OC variations and system topology changes.

Chapter 5 extends the effort made in Chapter 4 to develop a on-line DSA scheme that is robust to missing synchrophasor data. In on-line DSA, synchrophasor data can become unavailable due to the expected failure of PMUs or communication links. Similarly to the scheme of Chapter 4, the proposed scheme consists of three processing stages; the difference is that the random subspace technique is employed in the latter scheme. First, multiple small DTs are trained off-line, each by using a randomly selected attribute subset. In near real-time, new cases are used to test the small DTs. The test results are then utilized in on-line DSA to choose a few *viable* small DTs (i.e., the DTs without missing data from their attribute subsets) and calculate the voting weights via a boosting process. Finally, security classification decisions of on-line DSA are obtained via a weighted voting of the chosen small DTs. Specifically, the randomized algorithm for selecting attribute subsets exploits the locational information of attributes and the availability of synchrophasor data, so as to guarantee that a significant portion of small DTs are still viable when some synchrophasor data become missing. Further, the process of boosting viable small DTs in on-line DSA guarantees that accurate decisions can be obtained by making use of the viable small DTs, in the sense that the viable small DTs are iteratively chosen in a gradient descent manner and assigned proper voting weights, leading to the high robustness and accuracy of the proposed approach in case of missing synchrophasor data.

Chapter 6 discusses the fault detection and localization of line outage by using synchrophasor data. For large-scale power grids, this task is challenging, due to the

massive scale, system uncertainty and inevitable measurement errors. It is clear that deterministic approaches would not work well. With this insight, probabilistic graphical models are utilized to model the spatially correlated synchrophasor data, and statistical hypothesis testing method is utilized for the task of fault detection and localization. The contributions of this study are twofold. 1) Based on the DC power flow model, a Gaussian Markov random field (GMRF) model is developed for the phasor angles across the buses. Specifically, based on the stochastic characteristics of power flows, phasor angles are modeled as Gaussian random variables, which have a latent dependency graph governed by the interconnection structure of the power grid. The relationship between the partial correlations of phasor angles and the physical parameters of power systems is also revealed. 2) A decentralized networked data fusion algorithm for fault detection and localization is developed, by exploiting the multi-scale decomposition property of GMRF.

Finally, Chapter 7 discusses future research directions.

PART I

SPATIO-TEMPORAL ANALYSIS FOR WIND FARM GENERATION FORECAST AND INTEGRATION

Chapter 2

SPATIO-TEMPORAL ANALYSIS FOR SHORT-TERM WIND FARM GENERATION FORECAST

2.1 Introduction

A critical aspect in meeting the renewable portfolio standard (RPS) goals adopted by many states in the U.S. includes the integration of renewable energy sources, such as wind and solar [7]. Given the fact that the power outputs of wind turbines are highly dependent on wind speed, the power generation of a wind farm varies across multiple timescales of power system planning and operations. With increasing penetration into bulk power systems, wind generation has posed significant challenges for reliable system operations, because of its high variability and non-dispatchability [8]. Specifically, one key complication arises in terms of committing and dispatching conventional generation resources, when the short-term forecast of wind farm generation is not accurate. Currently, wind generation forecast for an individual wind farm typically has an error of 15% to 20% [6], in sharp contrast to the case of load forecast. When the actual wind generation is above the forecasted value, i.e., more conventional generation capacity has been committed than needed, it could result in less efficient set points for thermal units. In some cases, wind generation may need to be curtailed [9]. On the flip side of the coin, when the actual wind generation is less than the forecasted value, costly ancillary services and fast acting reserves have to be called upon to compensate for the deficit. Therefore, it is imperative to develop accurate forecast approaches for wind farm generation.

State-of-the-art short-term wind power forecast approaches include time-series models (e.g., auto-regressive models [10], Kalman filtering [11]), Markov chains [12, 13], and data mining [14, 15]. A comprehensive literature review on wind power

forecast can be found in [16] and [17]. Time-series models and data mining-based regression models, while being able to provide continuous-value wind power forecast, could suffer from high computational complexity. Compared to other forecast models, finite-state Markov chains strike a good balance between complexity and modeling accuracy. In particular, given the state space and historical data, the state transition matrix can be computed by using the maximum likelihood estimation technique [12]; when new data points are available online, it is also very easy to update the transition matrix. Generally, aggregate wind power data is a double-bounded non-Gaussian time series [18]. In reference [19], the logit transform is carried out as preprocessing, so that such a bounded time series can be studied by using auto-regressive models in a Gaussian framework. In this chapter, finite-state Markov chains are utilized to model the bounded wind power time series with a general probability distribution. It is worth noting that finite-state Markov chains inherently have bounded support, and the stationary distribution of a Markov chain can be general. Despite the appealing features of Markov chains, there is no systematic method to design the state space of Markov chains for wind power modeling. The proposed approach in this chapter addresses this issue by developing a general spatio-temporal analysis framework. An overview of the main contributions is presented below.

2.1.1 Summary of Main Results and Contributions

One key observation of this study is the wind farm spatial dynamics, i.e., *the power outputs of wind turbines within the same wind farm can be quite different, even if the wind turbines are of the same class and physically located close to each other.* The disparity in the power outputs of wind turbines may be due to the wake effect of wind speed, diverse terrain conditions, or other environmental effects. Motivated by this observation, graph-learning based spatial analysis is carried out to quantify the statistical distribution of *wind farm generation*, with rigorous characterization of

wind farm spatial dynamics. Then, time series analysis is applied to quantify the level crossing rate (LCR) of the wind farm's aggregate power output. Finite-state Markov chains are then constructed, with the state space and transition matrix designed to capture both the spatial and temporal dynamics of the wind farm's aggregate power output. Based on [66], the distributional forecast and the point forecast of wind farm generation are provided by using the Markov chains and ramp trend information. In this work, another finding of independent interest is that the tail probability of wind farm's aggregate power output exhibits a "power-law" decay with an exponential cut-off, where the power-law part has a much heavier tail than the Gaussian distribution. This indicates that one cannot simply apply the central limit theorem (CLT) to characterize the aggregate power output, because of the strong correlation across the power outputs of wind turbines within a wind farm.

The main contributions of this study are summarized below:

- A general spatio-temporal analysis framework is developed, in which the spatial and temporal dynamics of wind farm generation are characterized by analytically quantifying the statistical distribution and the LCR.
- Built on the results of spatio-temporal analysis, a systematic approach for designing the state space of the Markov chain is introduced.
- By modeling variable wind power as a Markov chain, stochastic unit commitment and economic dispatch problems can be studied by using Markovian state-space approaches instead of scenario-based approaches [20, 21]. Thus, the complexity induced by exponentially-growing scenarios of scenario-based approaches can be avoided. *Therefore, this study is a timely contribution to the recent efforts on wind generation integration that involves Markov-chain-based stochastic optimizations.*

The rest of this chapter is organized as follows. A few critical observations from the measurement data are first discussed in Section 2.2. Spatio-temporal analysis and the design of Markov chains are presented in Section 2.3. Section 2.4 discusses the proposed Markov-chain-based forecast approach and numerical examples. Conclusions are provided in Section 2.5.

2.2 Available Data and Key Observations

In this chapter, spatio-temporal analysis is carried out for a large wind farm with a rated capacity of $P_{ag}^{\max}$=300.5MW. There are $M = 2$ classes of wind turbines in this wind farm, with $N_1 = 53$ and $N_2 = 221$, respectively. The power curves of the two turbine classes are provided in Fig. 2.1. For each class C_m, a meteorological tower (MET) H_m is deployed and co-located with a wind turbine, denoted by r_m. The power outputs of all wind turbines and the wind speeds measured at all METs are recorded every 10 minutes for the years 2009 and 2010. From the measurement data, several key observations can be made as follows.

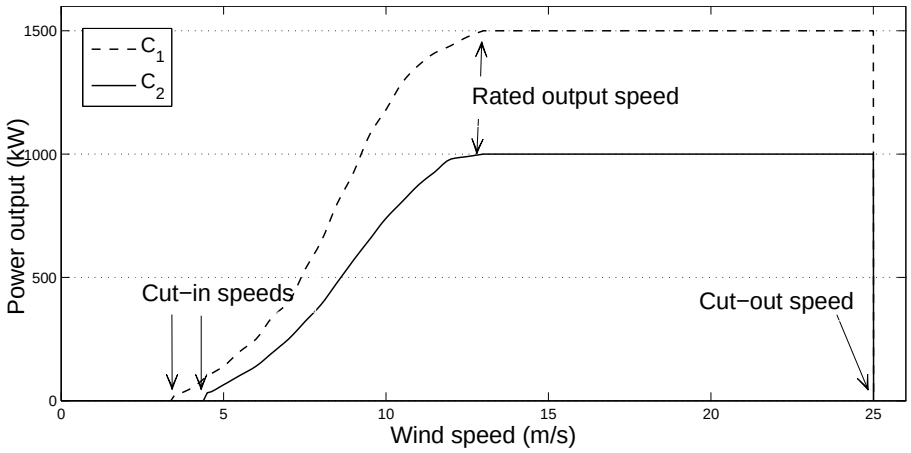


Figure 2.1: Power curves for wind turbines from classes C_1 and C_2.

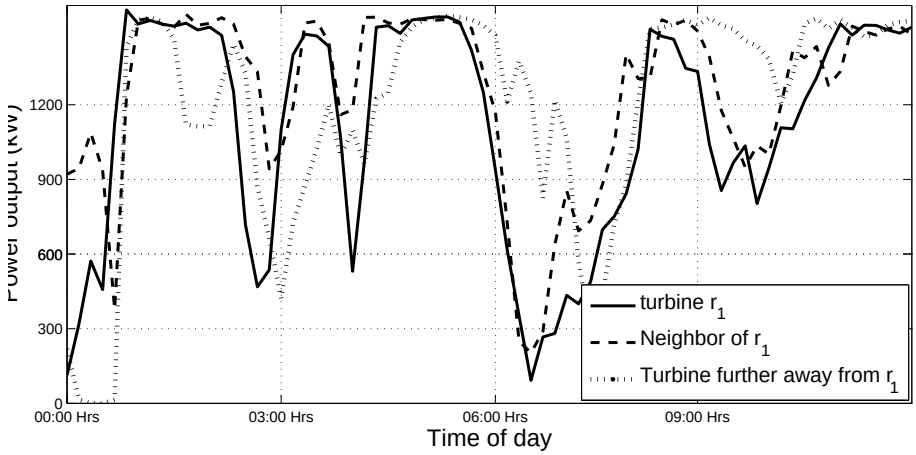


Figure 2.2: Power outputs of three wind turbines in C_1.

2.2.1 Spatial Dynamics of Wind Farm

A critical observation from the measurement data is that the power outputs of wind turbines within the wind farm can be quite different. Fig. 2.2 illustrates the power outputs of three wind turbines in C_1. It is clear that the power outputs are not equal, despite the geographic proximity of r_1 and its nearest neighbor (the disparity in the power outputs of the wind turbines belonging to C_2 has also been observed; the plots are not included for the sake of brevity). This disparity has been largely neglected in the existing literature.

Although the variable power outputs of wind turbines are not identical, it is reasonable to assume that they follow the same probability distribution if the wind turbines are of the same class. A natural question here is whether the CLT, either the classic CLT or the generalized CLT, can be applied to characterize the probability distribution of the aggregate power output of a large number of wind turbines. To this end, the tail probability distribution of the wind farm's aggregate power output is examined and plotted in Fig. 2.3. As illustrated in Fig. 2.3, the tail probability demonstrates a "power-law" decay with an exponential cut-off and the

power-law part has a much heavier tail than the Gaussian distribution. It is useful to note that this kind of tail behavior has been observed in many natural phenomena (e.g., size of forest fires) that have strong component-wise correlations [22]. Because of the strong correlation between the power outputs of wind turbines, particularly from adjacent wind turbines, the classic CLT cannot be applied to characterize the probability distribution of the wind farm's aggregate power output. In fact, even the "CLT under weak dependence" cannot be directly applied, despite the fact that the correlation between the power outputs of wind turbines weakens with the distance between them (the "mixing distance"). Hence, the probability distribution of the wind farm's aggregate power output cannot be characterized using the classic CLT; and it may not even be governed by stable laws [23]. With this insight, the proposed approach resorts to graphical learning methods to model the dependence structure in the power outputs of individual wind turbines and carries out spatio-temporal analysis accordingly.

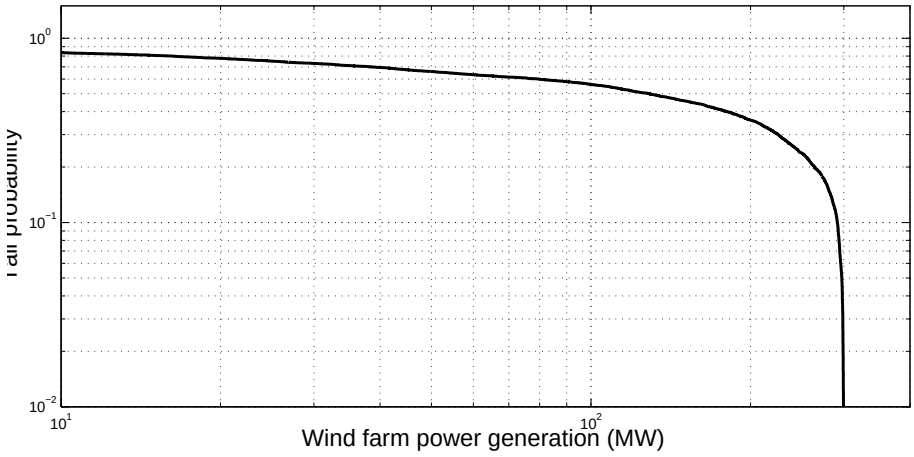


Figure 2.3: Tail probability of the wind farm's aggregate power output.

2.2.2 Diurnal Non-Stationarity and Seasonality

Another key observation, as illustrated in Fig. 2.4, is the diurnal non-stationarity and the seasonality of wind farm generation. Specifically, it is observed that within

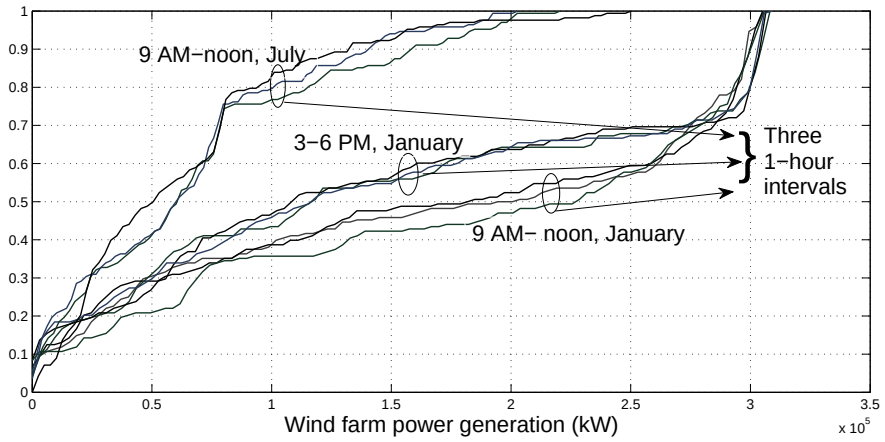


Figure 2.4: Empirical distribution of wind farm generation over various 1-hour intervals of different epochs of the day and different months.

each three-hour epoch, the probability distributions of wind farm generation over three consecutive 1-hour intervals are consistent. However, these CDFs from different epochs of three hours and different seasons can be quite different, indicating the non-staionarity of wind farm generation. The non-stationarity of probability distributions of wind farm generation caused by diurnal pattern and seasonality can be handled by developing a Markov chain for each epoch of the day and for each month.

In what follows, data-driven analysis is carried out to characterize the spatial and temporal dynamics of the wind farm's aggregate power output. The data for the year 2009 is used in spatio-temporal analysis to guide the design of Markov chains, and the data for the year 2010 is used to assess the accuracy of the forecast provided by the proposed Markov-chain-based approach. Specifically, the 9 AM-noon epoch of January 2009 is used as an illustrative example in the following spatio-temporal analysis, since *this epoch exhibits the richest spatio-temporal dynamics, in the sense that the wind farm's aggregate power output during this epoch takes values ranging from 0 to the wind farm's rated capacity and exhibits the highest variability over time (quantified by LCR).*

2.2.3 Weibull Distribution of Wind Speed

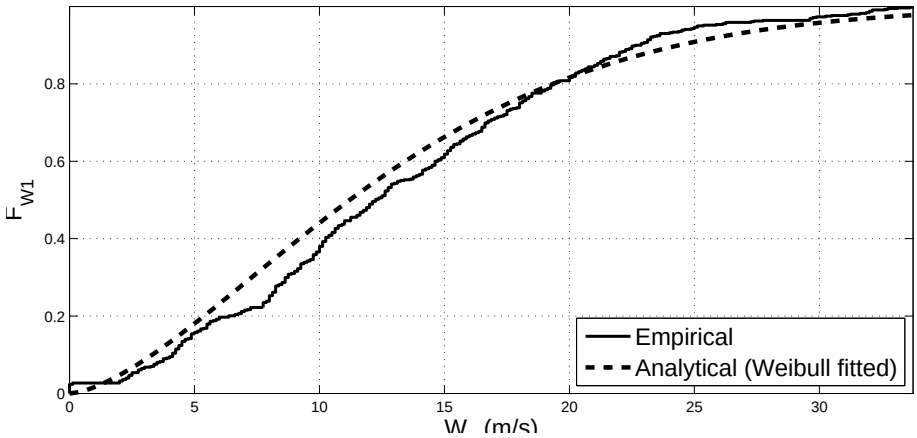


Figure 2.5: Weibull-fitted CDF (λ=11.37, k=1.54) and empirical CDF of W_1 for the 9 AM-noon epoch of January 2009.

In the existing literature, wind speed is usually characterized using Weibull distributions [24]. In this work, it is observed from the measurement data that the wind speed W_m at each MET within the wind farm closely follows a Weibull distribution during each epoch, the probability density function (PDF) of which is given by:

$$f_{W_m}(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp^{-(x/\lambda)^k}, \quad \forall x \geq 0, \tag{2.1}$$

where k is the shape parameter and λ is the scale parameter. Specifically, the maximum likelihood estimation technique [25] is utilized to estimate the parameters from the measurement data. The fitted cumulative density function (CDF) and the empirical CDF of W_1 for the 9 AM-noon epoch of January 2009 are plotted in Fig. 2.5. Note that reference [25] presents several other methods for estimating the parameters of the Weibull distribution, in which one method with good visual interpretation is the Weibull plot adopted by [26].

2.3 Spatio-temporal Analysis of Wind Farm Generation

2.3.1 Spatial Analysis and Statistical Characterization

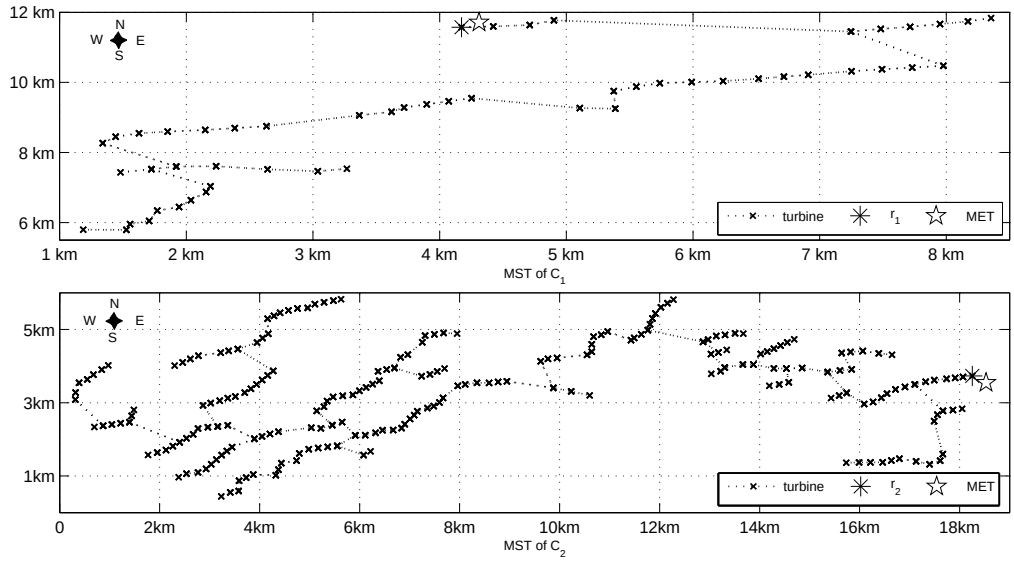


Figure 2.6: MST of C_1 (with distance to the southwest corner of the wind farm).

A key objective of spatial analysis is to characterize the statistical distribution of $P_{ag}(t)$. To this end, regression analysis is applied to the measurement data of each turbine's power output, so that $P_{ag}(t)$ could be expressed in terms of wind speed. Then, the analytical CDF of $P_{ag}(t)$ can be obtained from the fitted Weibull CDF of wind speed. In what follows, the key steps of spatial analysis are provided in detail.

Using the geographical information of wind turbines, an MST with r_m as the root node is constructed for each class C_m by using Prim's algorithm [27], as illustrated in Fig. 2.6. It is easy to see that for each wind turbine i in C_m, there exists only one path from r_m to i in the MST of C_m. Define the node which is closest to i along this path as the "parent" node of i. Another key observation from the measurement data is that an affine relationship exists between the parent-child turbine pairs for each class, with the case of C_1 illustrated in Fig. 2.7. Therefore, a

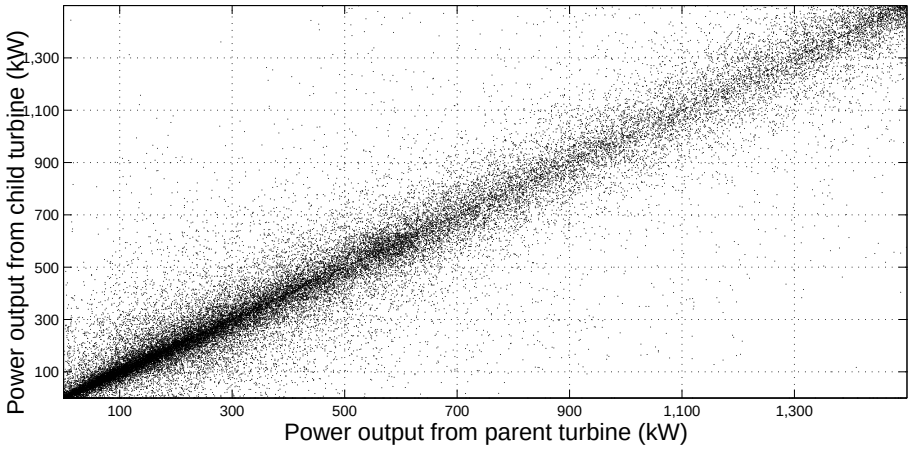


Figure 2.7: Power outputs of parent-child turbine pairs of C_1 for the 9 AM-noon epoch of January 2009.

coefficient α_m is introduced for C_m, and the linear regression model $P_k(t)=\alpha_m P_j(t)$ is used for each parent-child turbine pair (j,k) in C_m accordingly. Further, define $d_m(i)$ as the number of the nodes (excluding node i) along the path from r_m to node i, then the linear regression model $P_i(t)=\alpha_m^{d_m(i)} P_{r_m}(t)$ can be used for any wind turbine i in C_m. The value of α_m is determined by applying the minimum mean square error (MMSE) principle to the aggregate power output of C_m, as follows:

$$\alpha_m = \underset{\alpha}{\operatorname{argmin}} \frac{1}{N_t} \sum_t (P_{ag,m}(t) - \sum_{i \in C_m} \alpha^{d_m(i)} P_{r_m}(t))^2. \tag{2.2}$$

Similarly, an affine relationship between the wind speeds is also observed from the measurement data. For convenience, H_1 is chosen as the reference MET, i.e., $\overline{m}=1$. Then, the linear regression models for wind speeds are given by $W_m(t)=\beta_m W_{\overline{m}}(t)$, where β_m is solved using the MMSE principle as follows:

$$\beta_m = \underset{\beta}{\operatorname{argmin}} \frac{1}{N_t} \sum_t (W_m(t) - \beta W_{\overline{m}}(t))^2. \tag{2.3}$$

Using $P_{r_m}(t)=U_m(W_m(t))$, the aggregate power output of the wind farm could be characterized as follows:

$$P_{ag}(t) = \sum_m P_{ag,m}(t) = \sum_m \sum_{i \in C_m} \alpha_m^{d_m(i)} U_m(\beta_m W_{\overline{m}}(t))$$

$$\triangleq G_{pw}(W_{\overline{m}}(t)). \tag{2.4}$$

It is easy to see that due to the monotone characteristics of $U_m(\cdot)$, $G_{pw}(\cdot)$ is monotonically increasing. Therefore, the analytical CDF of $P_{ag}(t)$ can be obtained from the fitted Weibull distribution of $W_{\overline{m}}(t)$, given by $F_{P_{ag}}(\cdot) = F_{W_{\overline{m}}}(G_{pw}^{-1}(\cdot))$. The analytical CDF and the empirical CDF of $P_{ag}(t)$ for the considered epoch are illustrated in Fig. 2.8. Note that the discontinuity in the analytical CDF of $P_{ag}(t)$ is a result of the cut-out wind speed (25 m/s), as plotted in Fig. 2.1.

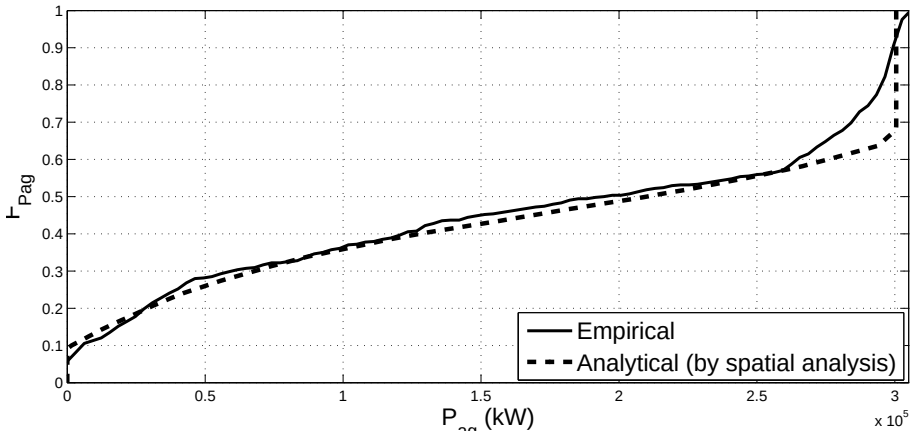


Figure 2.8: CDF of $P_{ag}(t)$ for the 9 AM-noon epoch of January 2009.

It is worth noting that the linear regression models with homogeneous regression coefficients used here are motivated by the observation from the measurement data. The above regression analysis could be generalized by applying more general regression analysis methods. For example, each parent-child turbine pair can have a different linear regression coefficient or the parent-child turbine pairs can be analyzed by using different regression models.

2.3.2 Temporal Analysis and LCR Quantification

During each epoch, both the wind speed $W_{\overline{m}}(t)$ and the wind farm generation $P_{ag}(t)$ could be regarded as stationary stochastic processes. The LCR of a stochastic process is formally defined as the number of instances per unit time that the stochastic process crosses a level in only the positive/negative direction [28]. Intuitively, $L_{P_{ag}}(\cdot)$ quantifies how frequently $P_{ag}(t)$ transits between different generation levels. It will be apparent soon that $L_{P_{ag}}(\cdot)$, together with the statistical characterization $F_{P_{ag}}(\cdot)$, is critical in designing the state space representation of the Markov chains used for wind farm generation forecast.

It is useful to note that due to the discontinuity in $F_{P_{ag}}(\cdot)$, as illustrated in Fig. 2.8, a smooth Gaussian transformation for $P_{ag}(t)$ is unattainable. Hence, the LCR of wind speed is first characterized. In order to quantify $L_{P_{ag}}(\cdot)$ analytically, $L_{W_{\overline{m}}}(\cdot)$ is first derived and converted to $L_{P_{ag}}(\cdot)$ by using the mapping defined in (2.4). To this end, autoregressive analysis is applied to $W_{\overline{m}}(t)$. As argued in [29], autoregressive analysis preceded by transforming the stationary non-Gaussian process $W_{\overline{m}}(t)$ to a Gaussian process can result in a better fit, compared with fitting to an autoregressive model directly. Therefore, $W_{\overline{m}}(t)$ is transformed to a standard normal random variable, given by

$$W_{\overline{m}}^{\mathcal{N}}(t) = F_{\mathcal{N}}^{-1}(F_{W_{\overline{m}}}(W_{\overline{m}}(t))), \tag{2.5}$$

A first-order autoregressive (AR(1)) model [30] is then fitted to $W_{\overline{m}}^{\mathcal{N}}(t)$:

$$W_{\overline{m}}^{\mathcal{N}}(t) = \phi W_{\overline{m}}^{\mathcal{N}}(t-1) + \epsilon(t), \tag{2.6}$$

where the white noise term is modeled as a zero-mean Gaussian random variable $\epsilon(t) \sim \mathcal{N}(0, \sigma_\epsilon^2)$. The parameters ϕ and σ_ϵ can be estimated by solving the Yule-Walker equations [30]. Then, the LCR of $W_{\overline{m}}^{\mathcal{N}}(t)$ for a specific wind speed level γ (γ>0) can be calculated using the following steps:

$$
\begin{aligned}
& L_{W_{\overline{m}}^{\mathcal{N}(0,1)}}(\gamma) \\
&= \int_{-\infty}^{\gamma} \Pr(W_{\overline{m}}^{\mathcal{N}(0,1)}(t) > \gamma | W_{\overline{m}}^{\mathcal{N}(0,1)}(t-1) = w) f_{\mathcal{N}(0,1)}(w) \mathrm{d}w \\
&= \int_{-\infty}^{\gamma} \Pr(\epsilon(t) > \gamma - \phi w) f_{\mathcal{N}(0,1)}(w) \mathrm{d}w \\
&= \int_{-\infty}^{\gamma} \left(1 - F_{\mathcal{N}(0,1)}\left(\frac{\gamma - \phi w}{\sigma_\epsilon}\right)\right) f_{\mathcal{N}(0,1)}(w) \mathrm{d}w. \qquad (2.7)
\end{aligned}
$$

It is easy to see that $L_{W_{\overline{m}}}(\cdot)$ can be obtained from $L_{W_{\overline{m}}^{\mathcal{N}(0,1)}}(\cdot)$ using the inverse mapping of the strictly increasing function defined in (2.5). Further, using the monotonically increasing function defined in (2.4), the LCR of $P_{ag}(t)$ for a specific wind farm generation level Γ ($\Gamma \in (0, P_{ag}^{\max}]$) is given by:

$$
L_{P_{ag}(t)}(\Gamma) = L_{W_{\overline{m}}^{\mathcal{N}(0,1)}}(F_{\mathcal{N}(0,1)}^{-1}(F_{W_{\overline{m}}}(G_{pw}^{-1}(\Gamma)))). \qquad (2.8)
$$

The procedure presented above completes the characterization of the analytical LCR of $P_{ag}(t)$ for an arbitrary epoch. The analytical LCR and the empirical LCR of $P_{ag}(t)$ for the 9 AM-noon epoch of January 2009 are illustrated in Fig. 2.9.

2.3.3 Markov Chain Model for Spatio-temporal Wind Power

A critical step in developing the Markov-chain-based forecast approach is to capture the statistical distribution and the temporal dynamics of $P_{ag}(t)$ during each epoch using a Markov chain with the following characteristics:

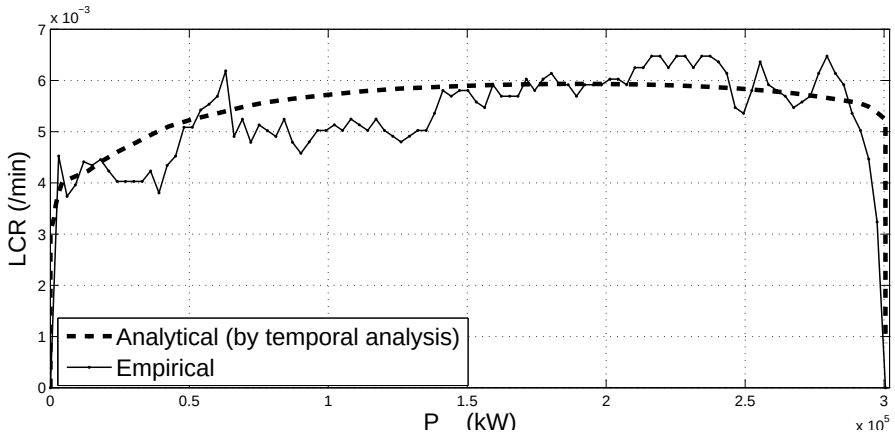


Figure 2.9: LCR of $P_{ag}(t)$ for the 9 AM-noon epoch of January 2009.

- The Markov chain has finite states. Specifically, state S_k (k=1,$\cdots$,N_s) corresponds to a specific range of generation levels [Γ_k,Γ_{k+1}), with Γ_1=0 and Γ_{N_s+1}=$P_{ag}^{\max}$.

- The Markov chain is discrete-time and of order 1.

The above characteristics are adopted to make the Markov chains practical for forecasting applications, so that forecast is made based on the most recent 10-min data only. But this can be generalized easily by using high-order Markov chains.

The procedure developed in [28] is utilized to design the state space. First, define τ_k as the average duration that $P_{ag}(t)$ falls in state S_k per unit time, given by:

$$\tau_k = \frac{F_{P_{ag}}(\Gamma_{k+1}) - F_{P_{ag}}(\Gamma_k)}{L_{P_{ag}}(\Gamma_{k+1}) + L_{P_{ag}}(\Gamma_k)}, \tag{2.9}$$

where $F_{P_{ag}}(\cdot)$ is the analytical CDF of $P_{ag}(t)$ that was characterized in spatial analysis, and $L_{P_{ag}}(\cdot)$ is the analytical LCR of $P_{ag}(t)$ derived in temporal analysis. By definition, τ_k quantifies the performance of the Markov chain in terms of how well the stochastic process $P_{ag}(t)$ is captured:

- A smaller value of τ_k suggests that $P_{ag}(t)$ is more likely to switch out of the state S_k within a 10-min slot, i.e., non-adjacent transitions are more likely to occur, *and hence the transitional behaviors of* $P_{ag}(t)$ *are not sufficiently captured by the discrete-time Markov chain.*

- If all the values of τ_k (k=1,$\cdots$,N_s) are too large, there would be fewer states, indicating that the quantization brought by the Markov chain is too crude, and the corresponding forecast would be less accurate.

Therefore, a key objective of state space design is to make each of τ_k (k=1,$\cdots$,N_s) fall into a reasonable range [28]. However, it is challenging to meet this design goal, especially when the closed-form expressions of $F_{P_{ag}}(\cdot)$ and $L_{P_{ag}}(\cdot)$ are unattainable. A practical solution adopted here is to introduce a constant τ and find the N_s-1 variables $\{\Gamma_2,\Gamma_3,\cdots,\Gamma_{N_s}\}$ by solving (2.9) numerically with $\tau_k=\tau$, $\forall k\in\{1,\cdots,N_s-1\}$. Once the state space $\mathcal{S}$ is designed, the transition probabilities could be easily estimated following the approach proposed in [12]. Specifically, the probability of a transition from S_i to S_j in a 10-min slot is given by

$$[Q]_{i,j} = \frac{n_{ij}}{\sum_{k=1}^{N_s} n_{ik}},\ i,j \in \{1,\cdots,N_s\}, \tag{2.10}$$

where $[\cdot]_{i,j}$ denote the entry at row i and column j of a matrix, and n_{ij} is the number of transitions from S_i to S_j encountered in the measurement data. The representative generation level for each state S_k, $k\in\{1,\cdots,N_s\}$, is determined using the MMSE principle, given by (the time index of $P_{ag}(t)$ is dropped for simplicity):

$$P_{ag,k} = \underset{P_k}{\operatorname{argmin}}\, \mathbf{E}_{P_{ag}}[(P_k - P_{ag})^2 | P_{ag} \in [\Gamma_k, \Gamma_{k+1})], \tag{2.11}$$

where $\mathbf{E}_{P_{ag}}[\cdot|P_{ag}\in[\Gamma_k,\Gamma_{k+1})]$ denotes taking expectation over P_{ag} conditioned on $P_{ag}\in[\Gamma_k,\Gamma_{k+1})$. Then, it is easy to see the representative generation level is given by:

$$P_{ag,k}=\frac{\int_{\Gamma_k}^{\Gamma_{k+1}} x f_{P_{ag}}(x)\mathrm{d}x}{F_{P_{ag}}(\Gamma_{k+1})-F_{P_{ag}}(\Gamma_k)}. \tag{2.12}$$

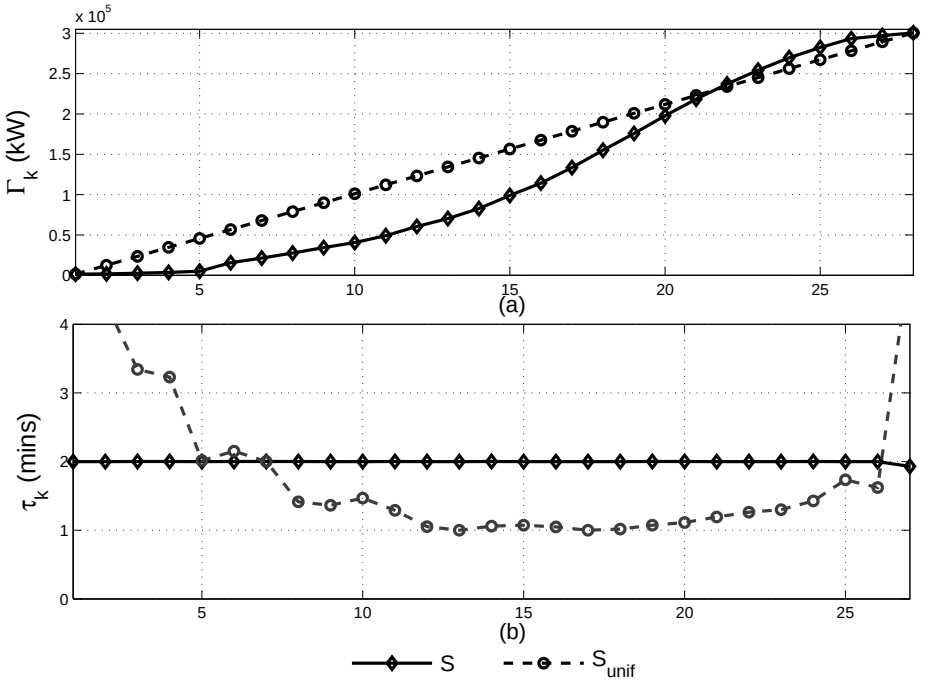


Figure 2.10: boundaries and average duration for each state of the Markov chain for the 9 AM-noon epoch of January 2009.

The above procedure is applied to the 9 AM-noon epoch of January 2009, by choosing τ=2 mins. The designed state space $\mathcal{S}$ and the values of τ_k are illustrated in Fig. 2.10, and the corresponding transition probabilities are plotted in Fig. 2.11. In the existing literature on Markov chain models for wind power [12] (not in the context of *wind farm* generation), all states of the Markov chain are simply chosen to be of uniform length. This simplistic design method is applied to the aggregate power output of the wind farm, by choosing $\Gamma_{k+1}=P_{ag}^{\max}k/N_s$, $\forall k\in\{1,\cdots,N_s-1\}$. The resultant state space, denoted by $\mathcal{S}_{\text{unif}}$, is compared with $\mathcal{S}$. From Fig. 2.10 and

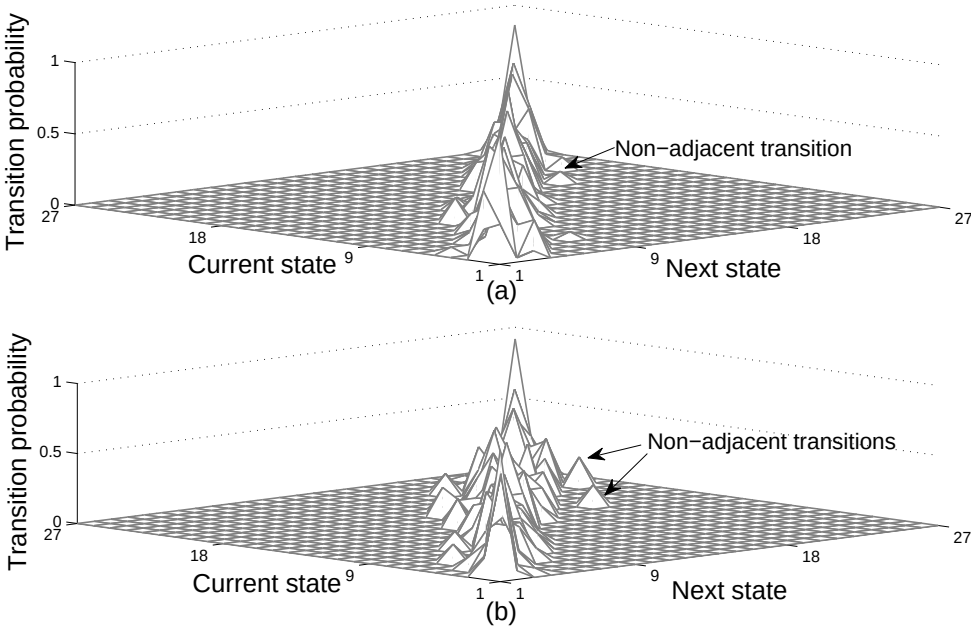


Figure 2.11: Transition matrix (a) by spatio-temporal analysis (b) by uniform quantization, for the 9 AM-noon epoch of January 2009.

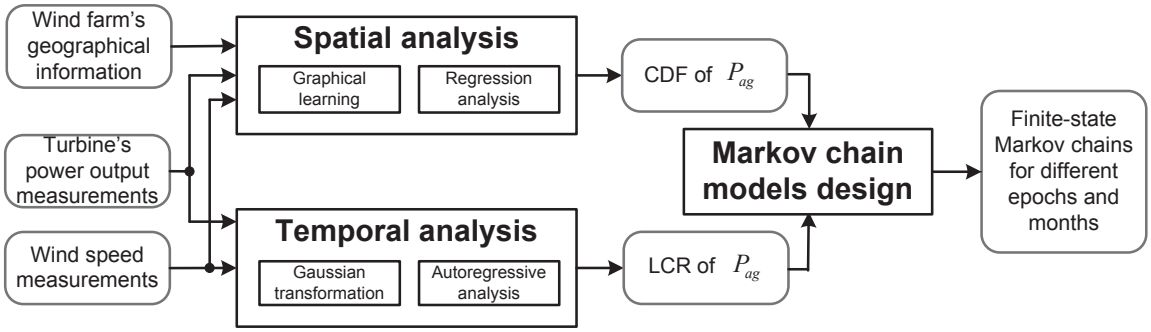


Figure 2.12: Offline spatio-temporal analysis (carried out for each epoch and each month by using historic measurement data).

Fig. 2.11, it is clear that higher values of τ_k are achieved for most of the states in $\mathcal{S}$, whereas more non-adjacent transitions are incurred by the Markov chain with state space $\mathcal{S}_{\text{unif}}$.

2.4 Markov-Chain-based Short-term Forecast of Wind Farm Generation

Given the current 10-min wind farm generation data $P_{ag}(t)$, the state of the Markov chain at time t, denoted by $S(t)$, is determined by searching for a state k_0 so that

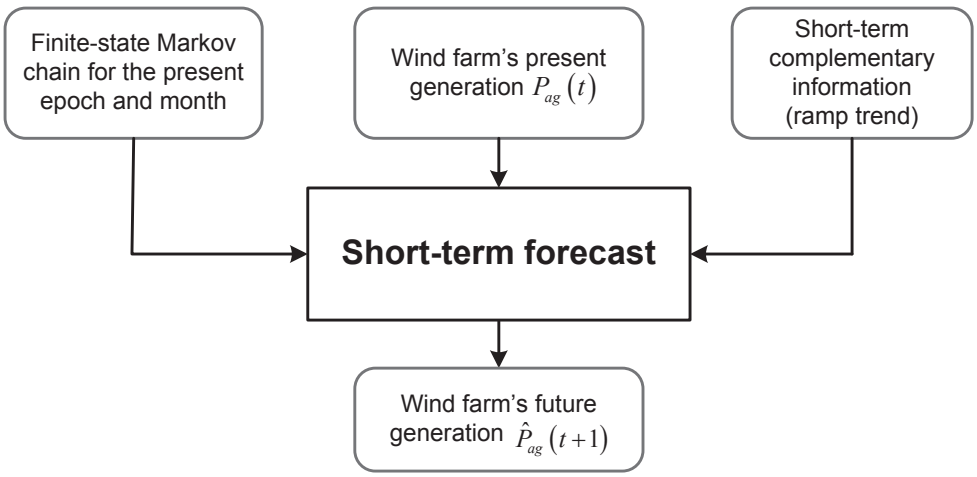


Figure 2.13: Online short-term forecasting.

$P_{ag}(t)\in[\Gamma_{k_0},\Gamma_{k_0+1})$. Thus, $S(t+1)$ and hence $P_{ag}(t+1)=P_{ag,S(t+1)}$ are random variables that depend on the transition matrix Q, $S(t)$ and $R(t)$. Further, let $R(t)=-1$ denote a decreasing trend, and $R(t)=1$ for the non-decreasing case. Then, the distributional forecast is given by

$$\Pr(P_{ag}(t+1) = P_{ag,j}|S(t), R(t)) =$$

$$\begin{cases} \dfrac{Q_{k_0,j}}{\sum\limits_{k\geq k_0}^{N_s} Q_{k_0,k}}, & \text{if } R(t) = 1 \text{ and } j \geq k_0 \\ \dfrac{Q_{k_0,j}}{\sum\limits_{k=1}^{k_0-1} Q_{k_0,k}}, & \text{if } R(t) = -1 \text{ and } j < k_0 \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

2.4.0.1 Point Forecasts

From the above distributional forecast, a point forecast can be derived by using the MMSE principle:

$$\hat{P}_{ag}(t+1) = \underset{P_{ag}}{\operatorname{argmin}}\, \mathbf{E}\left[(P_{ag} - P_{ag,S(t+1)})^2|S(t), R(t)\right] \tag{2.14}$$

Then, the solution to the above problem is given by:

$$\hat{P}_{ag}(t+1) = \begin{cases} \frac{\sum_{k \geq k_0}^{N_s} P_{ag,k} Q_{k_0,k}}{\sum_{k \geq k_0}^{N_s} Q_{k_0,k}}, & \text{if } R(t) = 1 \\ \frac{\sum_{k=1}^{k_0-1} P_{ag,k} Q_{k_0,k}}{\sum_{k=1}^{k_0-1} Q_{k_0,k}}, & \text{if } R(t) = -1 \end{cases} \tag{2.15}$$

which is exactly the mean value of the Markov chain conditioned on the currently observed state and the ramp trend.

2.4.1 Numerical Examples

2.4.1.1 Distributional Forecasts

By using the continuous rank probability score (CRPS) as a metric for distributional forecast, the *empirical skill* of the Markov-chain-based distributional forecast is defined as the average of the CRPS of the Markov-chain-based distributional forecast over all N_t realized data points, given by:

$$Skill = \frac{1}{N_t} \sum_t \int_0^{P_{ag}^{max}} \left(\hat{F}(x) - H(x - P_{ag}(t)) \right)^2 dx, \tag{2.16}$$

where $\hat{F}(x)$ is the CDF obtained by using the Markov-chain-based distributional forecast, and $H(x - P_{ag}(t)$ is the unit step function, which takes value 0 when $x<P_{ag}(t)$ and takes value 1 when $x \geq P_{ag}(t)$. Basically, *a higher skill value suggests that the Markov-chain-based distributional forecast is less accurate*. By using the above definition, the CRPS value of the Markov-chain-based distributional forecast over all the 52560 (365*24*6) data points of the year 2010 is calculated. The Skill of the Markov-chain-based distributional forecast over the data points of the year 2010 is provided in Table. 2.1. Since one main objective of this work is to develop Markov-chain-based distributional forecasting models, the Markov chain developed by the existing ap-

Table 2.1: Skill values of Markov-chain-based distributional forecasts over the testing data points of the year 2010.

	MC (unif.)	MC (τ=2)	MC (τ=1)
Skill	7.14 MW	6.27 MW	6.09 MW

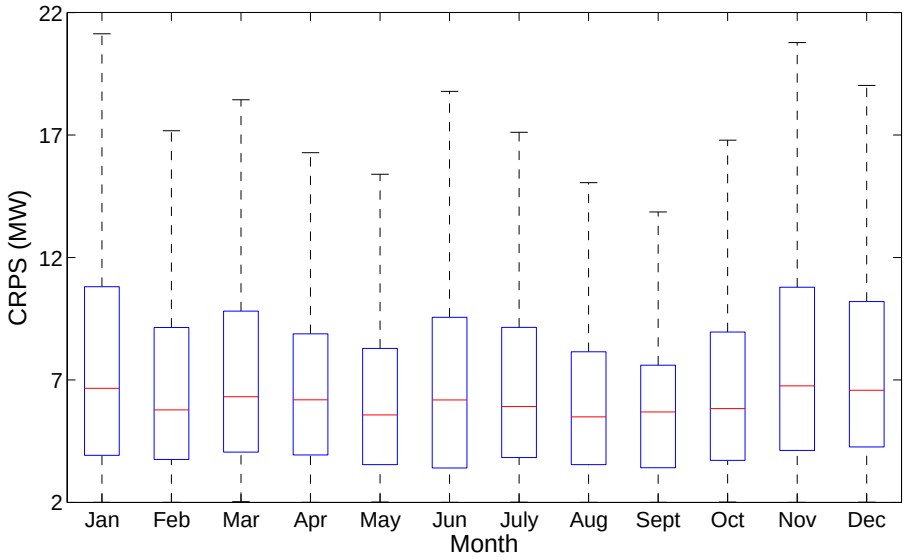


Figure 2.14: Statistics of CRPS over all months of the year 2010.

proach [12, 13] (uniform quantization) is used as a benchmark. The Markov chain developed by the proposed spatio-temporal analysis with the design parameter τ=2 (column 3 of Table. 2.1) has a skill score that is 13% less than that of the benchmark Markov chain that has the same number of states designed by uniform quantization (column 2 of Table. 2.1). By reducing the design parameter τ to 1, the forecasting performance of the Markov chain developed by the proposed spatio-temporal analysis (column 4 of Table. 2.1) is further improved.

To further examine the skill of the developed Markov-chain-based distributional forecasting method over different epochs and different month, the mean and percentiles of the CRPS values over the data points for each month or each epoch is computed. In the box plots of Fig. 2.14 and Fig. 2.15, the red bar in a box represents the mean value of the CRPS values over all data points that fall into a specific

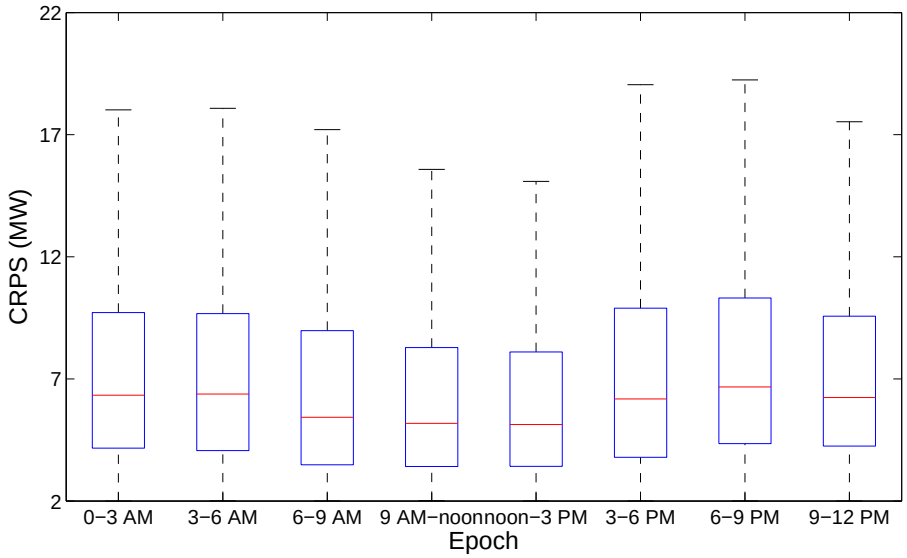


Figure 2.15: Statistics of CRPS over all 8 epochs of the year 2010.

epoch or a specific month. The top edge and bottom edge of a box represent the 25th and 75th percentiles, respectively. The top bar and bottom bar correspond to the extremes calculated from 1.5 interquartile ranges. It is observed from Fig. 2.15 that the means and standard deviations of the CRPS values are a little higher during afternoon-night epochs. Fig. 2.14 shows that the means of the CRPS values have little variability across different months, and the standard deviations of the CRPS values are slightly higher across the winter season. Another key observation from the results of numerical experiments is that *the CRPS of the Markov-chain-based distributional forecast over a realized data points $P_{ag}(t)$ is highly dependent on the ramp rate of $P_{ag}(t)$ at time t*. Here, the ramp rate of $P_{ag}(t)$ is defined as the absolute value of the change in the wind farm generation in a 10-min slot. For example, the ramp rate of $P_{ag}(t)$ at time t is given by $|P_{ag}(t)-P_{ag}(t-1)|$. By using the data points of the year 2010, the corresponding 52560 pairs of ramp rates and CRPS values are plotted in Fig. 2.16. It is observed that the ramp rates of $P_{ag}(t)$ and the CRPS values of the Markov-chain-based distributional forecast over a realized data points $P_{ag}(t)$ follows a

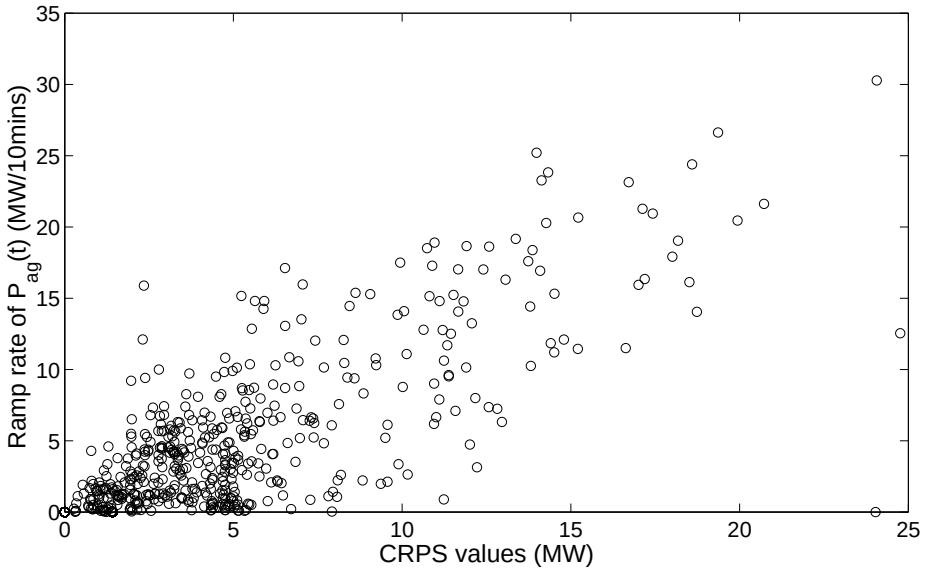


Figure 2.16: Correlation between the ramp rates of $P_{ag}(t)$ and the CRPS values of distributional forecast.

positive correlation. The above observation also explains the "phase transition" from the noon-3 PM epoch to the 3-6 PM in Fig. 2.15, i.e., the increased wind ramp caused by the sudden change in diurnal heating/vertical mixing conditions [31]. In summary, the statistics (especially the mean value) of the CRPS values vary slightly differently over different months and epochs, which suggests that the developed Markov-chain-based distributional forecasting methods deliver consistent forecasting performance across the entire year.

Further, three episodes of prediction intervals are plotted to better illustrate the developed Markov-chain-based distributional forecasts. According to the above observation, three representative time periods are chosen: 1) the 3-6 PM epoch of January 8th, 2) the 0-3 AM epoch of January 23rd, and 3) the 3-6 PM epoch of April 16th. The first period is chosen because January and the 3-6 PM epoch have the highest mean CRPS value (i.e., least accurate forecasts), and the CRPS value of January 8th is mostly close to the corresponding mean value. For the second period, January 23rd is an extreme day that had the highest average ramp rate among all

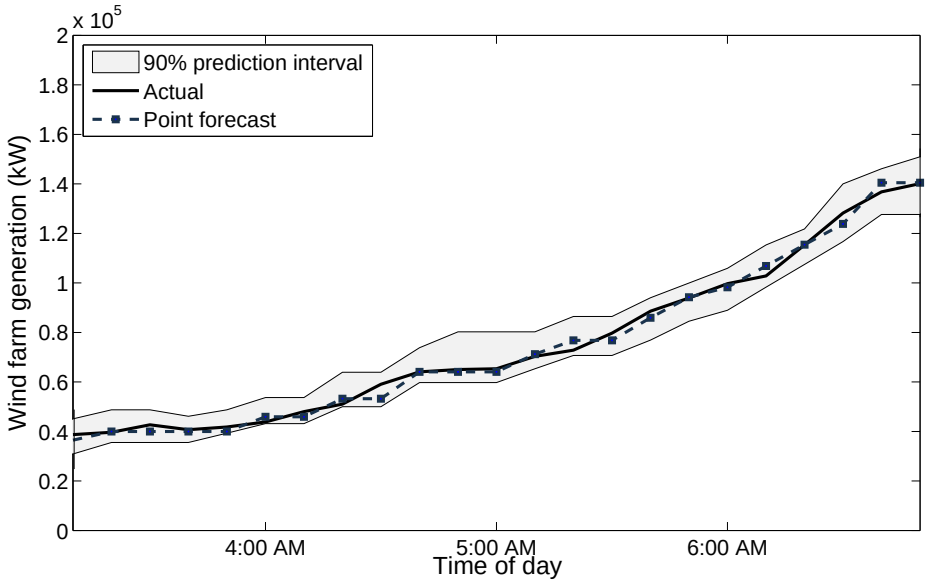


Figure 2.17: 10-min distributional forecasts on January 8th, 2010.

January days, and the 0-3 AM epoch experienced a large down-ramp from 75% to 25% of the rated capacity. The third period is chosen due to similar reasons as the first period, except that April is the month that has the least CRPS values. Fig. 2.17-2.19 illustrate the 90% prediction intervals obtained by the developed Markov-chain-based distributional forecasts. It is observed at for all three representative periods, the realized wind farm generation reasonably lies in the 90% prediction intervals.

2.4.1.2 Point Forecasts

By comparing the point forecast $\hat{P}_{ag}(t)$ with the actual wind farm generation $P_{ag}(t)$, forecast errors are quantified by mean absolute error (MAE), defined as

$$\text{MAE} = \frac{1}{N_t} \sum\nolimits_t |P_{ag}(t) - \hat{P}_{ag}(t)|, \tag{2.17}$$

mean absolute percentage error (MAPE), defined as

$$\text{MAPE} = \frac{\sum_t |P_{ag}(t) - \hat{P}_{ag}(t)|}{\sum_t P_{ag}(t)}, \tag{2.18}$$

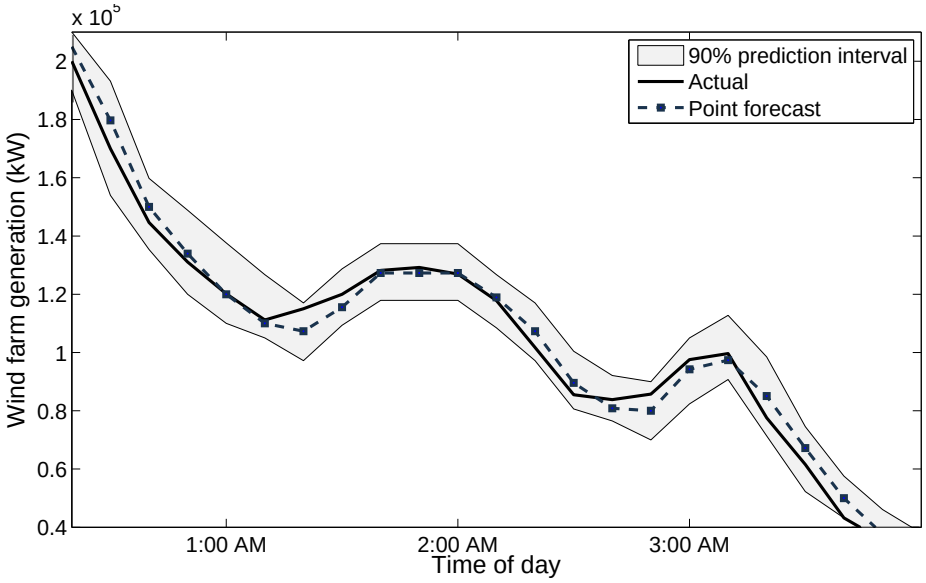


Figure 2.18: 10-min distributional forecasts on January 23rd, 2010.

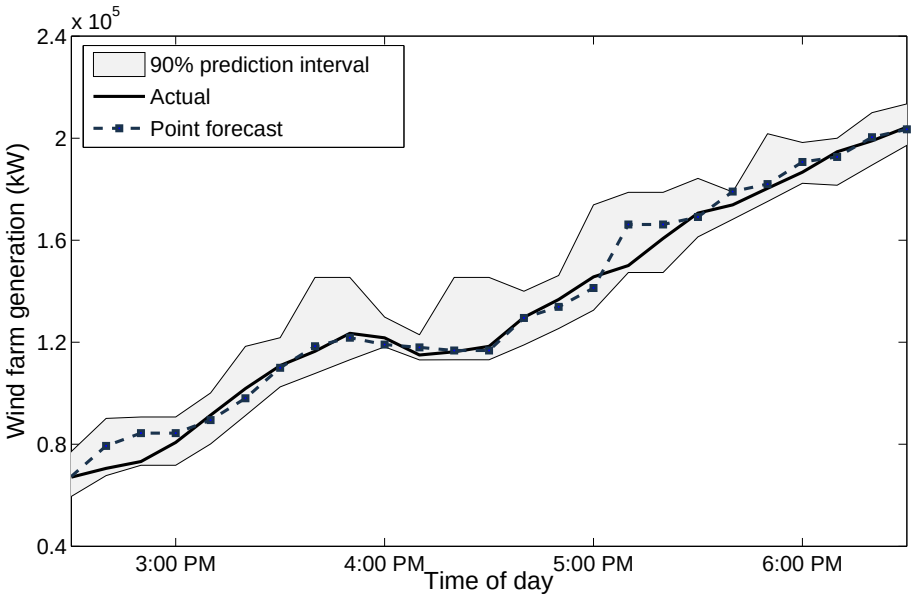


Figure 2.19: 10-min distributional forecasts on April 16th, 2010.

and root mean square error (RMSE), defined as

$$\text{RMSE} = \sqrt{\frac{\sum_t |P_{ag}(t) - \hat{P}_{ag}(t)|^2}{N_t}}. \tag{2.19}$$

Table 2.2: 10-min point forecast error of wind farm generation over the period shown in Fig.2.17.

Error	Persistent	MC (unif)	MC (τ=2)	MC (τ=1)
MAE	7.45 MW	7.61 MW	6.87 MW	6.71 MW
MAPE	8.75 %	8.94 %	8.07 %	7.88 %
RMSE	8.12 MW	8.24 MW	7.47 MW	7.31 MW

Table 2.3: 10-min point forecast error of wind farm generation over the period shown in Fig.2.18.

Error	Persistent	MC (unif)	MC (τ=2)	MC (τ=1)
MAE	9.63 MW	9.88 MW	7.12 MW	6.85 MW
MAPE	7.53 %	7.73 %	5.58 %	5.36 %
RMSE	10.13 MW	10.37 MW	7.54 MW	7.25 MW

Two point forecast approaches are used as benchmark:

- persistent forecast [32]: $\hat{P}_{ag}(t+1)$=$P_{ag}(t)$;
- forecast by Markov Chain with uniform quantization.

The test results by using the data for the three selected epoches and the year 2010 are provided in Table 2.2-2.5, respectively. It is observed that the proposed Markov-chain-based forecast approach has improved accuracy compared to the persistent forecast approach. *Note that the improvements for the ramp periods shown in Table. 2.2-2.4 are all higher than the average over the entire month in Table. 2.5.* Also note that the Markov chains based on uniform quantization give less accurate forecast than persistent forecast. This can be attributed to the uniform quantization not considering the spatio-temporal dynamics of wind farm generation.

Another key observation from Table 2.2 and Table 2.5 is that smaller values of τ leads to higher forecast accuracy of the Markov chains, at the cost of higher complexity of the Markov chains (in terms of the number of states). The trade-off

Table 2.4: 10-min point forecast error of wind farm generation over the period shown in Fig.2.19.

Error	Persistent	MC (unif)	MC (τ=2)	MC (τ=1)
MAE	8.43 MW	8.84 MW	6.92 MW	6.74 MW
MAPE	7.06 %	7.40 %	5.80 %	5.64 %
RMSE	8.87 MW	9.21 MW	7.23 MW	7.05 MW

Table 2.5: 10-min point forecast error of wind farm generation (all test data of the year 2010 is used).

Error	Persistent	MC (unif)	MC (τ=2)	MC (τ=1)
MAE	6.97 MW	7.14 MW	6.83 MW	6.62 MW
MAPE	6.68 %	6.84 %	6.55 %	6.34 %
RMSE	7.23 MW	7.54 MW	7.09 MW	6.89 MW

between the forecast accuracy and the complexity of the Markov chain for the 9 AM-noon epoch of January 2010 is illustrated in Fig. 2.20.

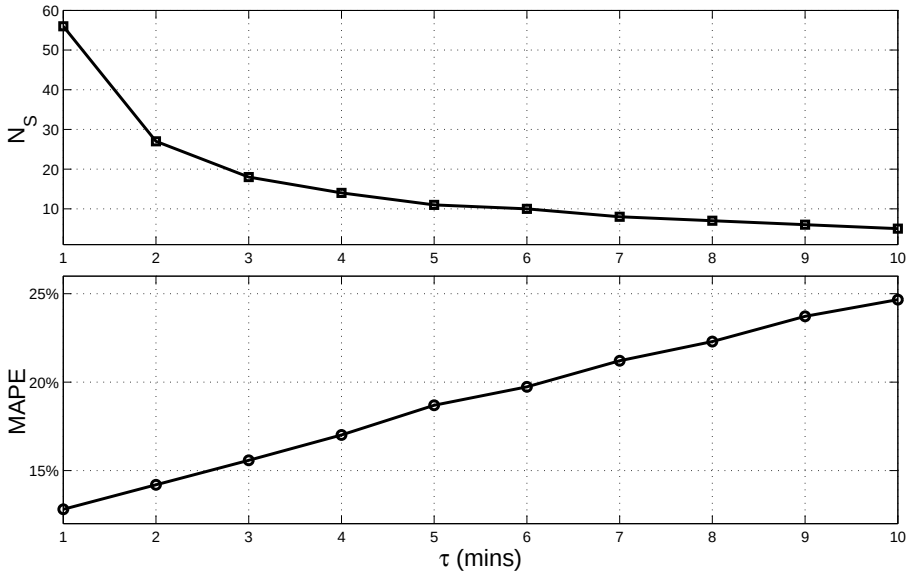


Figure 2.20: Number of states and the forecast error of Markov chains at various τ for the January 9 AM-noon epoch).

2.5 Conclusion

A general spatio-temporal analysis framework is developed for wind farm generation forecast, in which finite-state Markov chain models are derived. The state space, transition matrix and representative generation levels of the Markov chains are optimized by using a systematic approach. The short-term distributional forecast and point forecast are derived by using the Markov chains and the ramp trend information. One main contribution of this study is that the distributional forecast can be directly integrated into the problems of unit commitment and economic dispatch with uncertain wind generation, so that these problems can be studied in a general Markov-chain-based stochastic optimization framework. Further, the performance of the proposed Markov-chain-based point forecast approach is evaluated via numerical tests. Improved forecast accuracy of the point forecast over persistent forecast is observed.

In a related work [20], we are investigating power system economic dispatch with wind farm generation by utilizing a realistic test system and the Markov-chain-based distributional forecasts of wind farm generation. The distributional forecasts of wind farm generation are integrated into a stochastic programming framework of multi-period economic dispatch, so as to optimize the dispatch decisions over the operating horizon. The impact of the forecast errors of wind farm generation on economic dispatch is also studied.

Chapter 3

MULTI-TIMESCALE SCHEDULING WITH STOCHASTIC WIND GENERATION AND OPPORTUNISTIC DEMAND

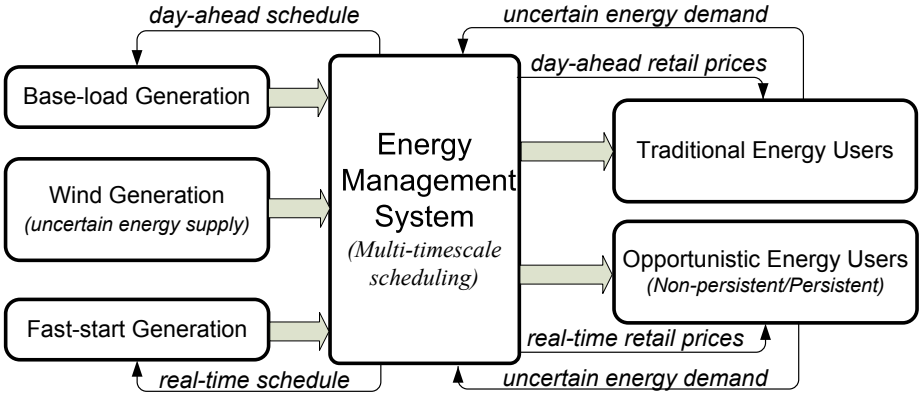
3.1 Introduction

Wind energy is expected to constitute a significant portion of all renewable generation being integrated to the bulk power grids of North America [5]. High penetration of wind generation not only brings many benefits, economically and environmentally, but also puts forth great operational challenges [33, 34]. Unlike conventional energy resources, wind generation is non-dispatchable, in the sense that wind energy could be not harvested simply by request. Further, wind generation highly depends on geographical and meteorological conditions and thus exhibits greater variability across all timescales, which makes it challenging for system operators to obtain accurate knowledge of future wind generation. Therefore, volatile and uncertain wind generation would have significant impact on the reliability of power systems, since the precise balance between the energy supply and demand at all time is of paramount significance to the reliable operations of power systems.

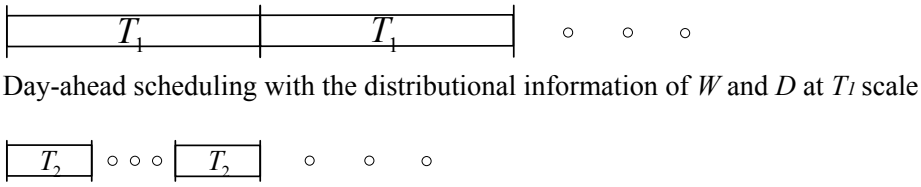
Recently, a significant amount of effort (e.g., [35–38]) has been directed towards integrating wind generation into the operation and planning of bulk power grids, in which wind generation is usually treated as negative load, and auto-regressive models (e.g., in [35]) or scenario trees (e.g., in [36,37]) are used to characterize the uncertainty in the net load. To cope with the uncertainty in the net load, the approaches proposed in [36–38] resort to the operating reserve (the additional generation capacities from online or fast-start generators) which is co-scheduled with the energy supply. However, as pointed out in [5], the variability and uncertainty in the demand is to a much lesser extent to that of wind generation, and hence, the cost of operating reserve

would increases significantly when the penetration level of wind generation is high. Towards high penetration levels of wind generation, changes will be required to the traditional methods used by system operators in order to maintain the reliability of bulk power grids.

With the objectives to maintain the reliability of bulk power grids with wind generation integration and reduce the operational cost involved, we observe that operating reserve could be obtained from the demand response of an emerging class of energy users, namely *opportunistic energy users*, instead of conventional generation. It is noted in [39] that over 10% daily electricity consumption in U.S. is from (residential and small commercial) energy users such as water heater, cloth dryers, and dish washers. Traditionally, these energy users pay a fixed price per unit of electricity that is established to represent an average cost of power generation over a given time-frame (e.g., a season). In smart grids with two-way communications, real-time pricing programs could be implemented so that the prices are tied with generation cost and vary according to the availability of energy supplies. In this scenario, these energy users would become *smart* by receiving and responding to real-time price signals, and are branded as opportunistic energy users, with the following behaviors distinct from *traditional energy users*: 1) they access the smart grid systems in an opportunistic manner, according to the availability of energy supply; 2) different from the "always-on" demand of traditional energy users, the load profiles of opportunistic energy users can be bursty and can be either inelastic or elastic; 3) the demand of opportunistic energy users respond to the price signals on a much finer timescale, and thus could be used to tune the balance between the energy supply and the demand in a real-time manner (i.e., within minutes). The prevalence of the new class of opportunistic energy users, if utilized intelligently, makes DSM a promising solution to reduce the costs incurred by high penetration of wind generation [40].



(a) Multi-timescale scheduling framework



(b) Scheduling horizon and timescales

Figure 3.1: A multi-timescale scheduling framework with integrated wind generation and two classes of energy users

3.2 A Multi-timescale Scheduling Framework

In this paper, we study multi-timescale scheduling in a framework as illustrated in Fig. 1. Specifically, the system operator procures energy supply from conventional generation and wind generation, and manage the demand of both classes of energy users via day-ahead/real-time prices to achieve system-wise reliability. Conventional generation, in turn, is drawn from two sources: base-load generators (e.g., thermal) and fast-start generators (e.g., gas turbines), in which base-load generators have stringent requirements on the minimum on/off time and ramp rates, and thus are scheduled one-day ahead and on hourly basis. The energy supply procurement and end-user pricing are performed in two stages, i.e., day-ahead scheduling and real-time scheduling, at different timescales. In day-ahead scheduling, with the distributional

information of wind generation W and energy demands D_t, the system operator decides the energy supply procurement s from base-load generators and the day-ahead price u, for the next day. In real-time scheduling, upon the realization of W and D_t, the system operator decides the real-time price v for opportunistic energy users, and uses fast-start generation or cancels part of scheduled base-load generation, as needed, to close the gap between demand and supply.

It is worth noting that the above framework that consists of day-ahead and real-time scheduling is developed based on the state-of-the-art scheduling schemes of power systems (e.g. [33,34,41]), by explicitly incorporating *opportunistic energy users* and the heterogeneous demand response of two classes of energy users.

3.2.1 General Problem Formulation

As illustrated in Fig. 3.1(b), a 24-hour period is divided into M T_1-slots of equal length, and each T_1-slot, in turn, consists of K T_2-slots. A T_1-slot and a T_2-slot can span an hour and 10 minutes, respectively. The objective of the system operator is to find a policy π that dictates the multi-timescale decisions s, u and v, so that the overall expected profit across the next day is maximized. Here, the scheduling problem is investigated from the perspective of an (vertically integrated) utility which owns the wind generation asset. Accordingly, wind generation is utilized as much as possible (with no curtailment). and the study could be generalized to deregulated markets where social welfare is maximized at the discretion of ISO. A general formulation of the multi-timescale scheduling problem is provided below:

$$\mathcal{P}: \quad \max_{\pi} \sum_{m=1}^{M} R_m^u(\pi), \tag{3.1}$$

where $R^u_m(\pi)$ is the total net profit in a T_1-slot, under policy π, given by[1]:

$$R^u_m(\pi) = \sum_{k=1}^{K} \mathbf{E}^{\pi}_{\psi^l_{k,m}} R^l_{k,m}(\psi^l_{k,m}, \pi), \tag{3.2}$$

where $R^l_{k,m}$ is the net profit in the kth T_2-slot of the mth T_1-slot (henceforth called the (k,m)th slot), and $\psi^l_{k,m}$ is the system state in the (k,m)th slot that is *observable* in real time[2]. When the opportunistic energy users are non-persistent, the system state consists of the wind generation and the energy demand from traditional users, i.e., $\psi^l_{k,m} = \{W_{k,m}, D_{t_{k,m}}\}$. When the opportunistic energy users are persistent, the system state is $\psi^l_{k,m} = \{W_{k,m}, D_{t_{k,m}}, P^l_{k,m}\}$, where $P^l_{k,m}$ denotes the number of persistent opportunistic energy users carried over from the previous T_2-slot to the (k,m)th slot.

3.2.2 Energy Supply and Demand Models

3.2.2.1 Stochastic wind generation

Development of wind generation forecast techniques has been the focus of a significant amount of research and industrial effort; see [6] for a detail discussion of wind generation forecasting in practical power systems. One insight revealed by the survey [6] is that integrating a significant amount of wind will largely depend on the accuracy of the wind power forecast. Therefore, by following the modeling method in [36], we model forecast error as a random variable. Thus, given a "point-forecast" $\hat{W}_m$, the wind generation amount in (k,m)th slot hat is known to the system operator in day-ahead scheduling is given by

$$W_{k,m} = \hat{W}_m + \varepsilon_{w_m}, \tag{3.3}$$

[1]The notation $\mathbf{E}^y_x$ denotes the expectation over x conditioned on y.

[2]The super-scripts "u" and "l" are used to distinguish between the upper-level quantities in a T_1-slot and the lower-level quantities in a T_2-slot.

where forecast error ε_{w_m} is closely tied to forecasting techniques, and hence can have arbitrary probability distributions and can also be non-stationary across T_1 slots. Another observation from [6] is that the variance of ε_{w_m} can be quite large (e.g., the mean absolute percentage error can be as high as 20%) for existing forecasting techniques in practice.

3.2.2.2 Generation cost

To facilitate qualitative analysis, we adopt a linear generation cost model derived from literatures [34,41]. Specifically, base-load generation incurs a cost of c_1 per unit when dispatched in real-time; fast-start generation incurs a cost c_2 per unit when dispatched in real-time as non-spinning reserves; base-load generation, scheduled day-ahead but canceled in real-time, incur a cost of c_p per unit. Typically, $c_2 > c_1 > c_p$. Further, we assume that wind generation is cost-free and utilized as much as possible.

3.2.2.3 Uncertain demand of traditional energy users

Based on [42], we model the energy demand of traditional energy users in the (k, m)th slot as a random variable with mean depending on the day-ahead price u_m, i.e.,

$$D_{t_{k,m}} = \alpha_{t_m} {u_m}^{\gamma_t} + \varepsilon_t, \tag{3.4}$$

where ε_t is a zero-mean random variable which accounts for the uncertainty of the demand of traditional energy users, and γ_t is the price elasticity which characterizes the price response of traditional energy users, and α_{t_m} is the normalizing constant. The price elasticity γ of energy users (either traditional or opportunistic), is defined in [42] as the ratio of percentage change in the expected demand to that of price, e.g., for the case of traditional energy users,

$$\gamma_t = \frac{u_m}{\mathbf{E}[D_{t_m}]} \frac{d\mathbf{E}[D_{t_m}]}{du_m}. \tag{3.5}$$

It is worth noting that the price elasticity is typically negative.

3.2.2.4 Uncertain demand of opportunistic energy users

Under real-time pricing, we assume that opportunistic energy users have the following behaviors:

- Opportunistic energy users arrive independently according to a Poisson process with rate λ_o, which is constant within a T_1-slot but can vary across the T_1-slots;
- In each T_2-slot, an opportunistic energy user i in the system decides to accept or reject the announced real-time price v by comparing with a price acceptance level V_i, which is randomly chosen and is $i.i.d$ across the opportunistic energy users.
- The expected demand of opportunistic energy users has a price elasticity γ_o, which is defined in a similar manner to that of traditional energy users in (3.5);
- Each active opportunistic energy user has a per-unit power consumption of E_o.

3.2.2.5 Day-Ahead and Real-Time Pricing

We consider the following multi-timescale end-user pricing model:

- Traditional and opportunistic energy users have separate contracts: traditional energy users pay day-ahead prices, and opportunistic energy users pay real-time prices;
- Traditional energy users are informed, one day ahead, of the day-ahead prices u corresponding to each T_1-slot;

- Opportunistic energy users receive the real-time price v at the beginning of each T_2-slot;

- *Both* day-ahead and real-time prices have price cap u_{cap} and v_{cap}, respectively. The price caps are motivated by the study [43] and intended to protect energy users by hedging against the risk under variable pricing.

3.2.3 Net Profit in a T_2-slot

Given the amount of conventional energy supply procurement s_m and the day-ahead price u_m settled for each T_2-slot of the m-th T_1-slot in day-ahead scheduling, together with the realizations of wind generation $W_{k,m}$ and traditional energy demand $D_{t_{k,m}}$, the net profit in the (k,m)th slot is

$$\begin{aligned} R^l_{k,m}(\psi^l_{k,m},\pi) = u_m D_{t_{k,m}} + \mathbf{E}^{\pi}_{D_{o_{k,m}}}[v_{k,m} D_{o_{k,m}} + (-c_p s_m)\mathbf{1}_A \\ + (-c_p \epsilon_{k,m} - c_1(s_m - \epsilon_{k,m}))\mathbf{1}_B \\ + (-c_1 s_m + c_2 \epsilon_{k,m})\mathbf{1}_C], \end{aligned} \tag{3.6}$$

where $D_{o_{k,m}}$ denotes the energy demand of opportunistic energy users, and the quantity $\epsilon_{k,m}$ is given by $\epsilon_{k,m}$=$W_{k,m}$+s_m−($D_{t_{k,m}}$+$D_{o_{k,m}}$). Indicator $\mathbf{1}_A$ corresponds to the scenario when the wind generation is sufficient to meet the demand of both classes of energy users. Indicator $\mathbf{1}_B$ refers to the scenario when the wind generation is not sufficient but there is energy supply surplus, which necessitates the cancelation of part of the scheduled base-load generation. Indicator $\mathbf{1}_C$ corresponds to the scenario when there is energy supply deficit and fast-start generation is necessary to close the gap. Formally, the indicators are described as follows:

$$\mathbf{1}_A = \begin{cases} 1 & \text{if } W_{k,m} \geq D_{t_{k,m}} + D_{o_{k,m}} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{1}_B = \begin{cases} 1 & \text{if } \mathbf{1}_A = 0 \text{ and } \epsilon_{k,m} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.7}$$

$$\mathbf{1}_C = \begin{cases} 1 & \text{if } \epsilon_{k,m} < 0 \\ 0 & \text{otherwise} \end{cases}$$

3.3 Multi-timescale Scheduling with Non-persistent Opportunistic Energy Users

The tight coupling between the decisions, in the sense that the energy supply procurement and prices in day-ahead scheduling have significant impact on real-time prices, and the real-time pricing policy, in turn, affect the realized net profit of day-ahead schedule, underscores the need for joint optimization of the day-ahead and real-time schedules. To this end, we take a "bottom-up" approach in solving the multi-timescale scheduling problem. Specifically, we first solve the real-time scheduling problem, conditioned on day-ahead decisions. Then, we investigate the day-ahead scheduling problem by taking into account the real-time pricing policy.

3.3.1 Real-time Scheduling on Timescale T_2

Recall that non-persistent opportunistic energy users response to high real-time price v by leaving the system, thus the total demand in the T_2-slot only depends on current decisions. Then, it suffices to examine the scheduling problem in a T_2-slot, which is formulated as[3]:

$$\mathcal{P}^{\mathbf{RT}}_{\mathbf{non-pst}} : \max_{v} R^l(\psi^l, s, u, v), \tag{3.8}$$

[3]We drop the suffix (k, m) for notational simplicity in this section.

where recall that ψ^l=$\{W, D_t\}$ is the system state, and (s, u) are the decisions made in day-ahead scheduling. The optimal solution to $\mathcal{P}^{\mathbf{RT}}_{\mathbf{non-pst}}$ defines a real-time pricing policy $\vartheta_{s,u}$:$\psi^l \rightarrow v$, a mapping from the system state to a real-time price conditioned on the day-ahead decisions (s, u).

3.3.2 Day-ahead Scheduling on Timescale T_1

The day-ahead scheduling problem could be re-formulated by taking into account the real-time pricing policy $\vartheta_{s,u}$. Further, since W and D_t are independent and identically distributed across the T_2-slots of the T_1-slot, the day-ahead scheduling problem can be optimized by simply considering the snapshot problem in a specific T_2-slot, given by:

$$\mathcal{P}^{\mathbf{DA}}_{\mathbf{non-pst}} : \max_{s,u} \mathbf{E}^u_{\psi^l} \left[R^l(\psi^l, s, u, \vartheta_{s,u}(\psi^l)) \right]. \tag{3.9}$$

3.3.3 Approximate Solutions

It is easy to see that a closed-form expression of $\vartheta_{s,u}$ in terms of arbitrary (s, u) is unattainable. This observation, along with the convolved nature of the uncertainties involved, makes a direct joint optimization of $\mathcal{P}^{\mathbf{DA}}_{\mathbf{non-pst}}$ challenging. We therefore take an alternative approach and obtain approximate solutions to the multi-timescale scheduling problems. In light of the characteristics of practical power systems, we impose the following conditions.

- **Condition I**: Wind generation is not sufficient to meet the total energy demand in the system.
- **Condition II:** The uncertainty in the demand of opportunistic energy users is significantly less than the uncertainty of wind generation.

Condition I is motivated by the Renewable Portfolio Standards of U.S. [44], in which most of the current state-by-state projected penetration levels of renewable generations (including wind generation) are below 30%. Under such a penetration level, it is unlikely that wind generation is sufficient to meet the overall load. We then provide an explanation of **Condition II**. Since opportunistic energy users arrive according to a Poisson process with rate λ_o, it follows that the number of active users which accept the current price v, denoted as N_a, is a Poisson random variable with mean $\lambda_o T_2 \mathbf{P}(V \geq v)$. Further, it is known that $\lambda_o T_2$ is typically large, and hence, N_a could be approximated by a Gaussian random variable. Note that the demand of the opportunistic energy users is given by $D_o = N_a E_o$, and recall that the price elasticity defined in (3.5):

$$\gamma_o = \frac{v}{\mathbf{E}[D_o]} \frac{d\mathbf{E}[D_o]}{dv}. \tag{3.10}$$

Then, it follows from (3.5) that

$$\mathbf{P}(V \geq v) \approx \alpha_o v^{\gamma_o}, \tag{3.11}$$

where $\alpha_o \triangleq v_{min}^{-\gamma_o}$ is a normalizing constant, and v_{min} denotes the highest price that is acceptable to all opportunistic energy users. Therefore, the demand of opportunistic energy users has a Gaussian distribution $\mathcal{N}(q_o(v), \sigma_o^2(v))$, with

$$q_o(v) \triangleq \lambda_o T_2 \alpha_o v^{\gamma_o} E_o,$$

$$\sigma_o^2(v) \triangleq \lambda_o T_2 \alpha_o v^{\gamma_o} E_o^2. \tag{3.12}$$

Observe from (3.12) that the variance of the demand of opportunistic energy users is of the same order as its mean. Further, wind generation and the demand of opportunistic energy users are typically comparable regarding the mean, and it is observed that the

uncertainty of wind generation is so high that the standard deviation is of the same order as its mean. Therefore, we conclude that **Condition II** holds.

3.3.3.1 Approximate Solution to the Real-time Scheduling Problem

It is easy to verify that $1_A = 0$ holds under **Condition I**. Then, by using (3.12), (3.6) reduces to

$$\begin{aligned}\tilde{R}^l(\psi^l, s, u, v) = uD_t - c_1 s + c_2 Y + (v - c_2)\, q_o(v) \\ - (2\pi)^{-1/2} c\sigma_o(v) \exp\left(-y^2/2\right) \\ - c\,(Y - q_o(v))\,(1 - Q(y))\,, \end{aligned} \tag{3.13}$$

where $Y \triangleq s+W-D_t$, $y \triangleq \frac{Y-q_o(v)}{\sigma_o(v)}$, $c \triangleq c_p-c_1+c_2$, and $Q(\cdot)$ is the tail probability of the standard normal distribution.

The demand of opportunistic energy users is said to be *relatively inelastic* if $-1 \leq \gamma_o < 0$, i.e., the percentage change in demand is greater than that of price; otherwise, it is *relatively elastic*. Since the price elasticity can have significant impact on the demand of opportunistic energy users, we proceed to study real-time schedule for different cases of elasticity.

Proposition 3.3.1. *Suppose* ***Condition I*** *and* ***Condition II*** *hold. When the demand of non-persistent opportunistic energy users is relatively inelastic, i.e.,* $-1 \leq \gamma_o < 0$*, the real-time pricing policy is given by* $\tilde{\vartheta}_{s,u}(\psi^l) = v_{cap}$.

Remarks: The proof is provided in Appendix A. Note that the above result is intuitive, since, with the opportunistic energy users' energy demand being relatively insensitive (inelastic) to the real-time price, the system operator can maximize profit by simply announcing the highest possible price, v_{cap}.

Proposition 3.3.2. *Suppose* ***Condition I*** *and* ***Condition II*** *hold. When the demand of non-persistent opportunistic energy users is relatively elastic, i.e.,* $\gamma_o < -1$*, the real-time pricing policy* $\tilde{\vartheta}_{s,u}$ *which maximize* $\hat{R}^l(\psi^l, s, u, v)$ *is given by*

$$\tilde{\vartheta}_{s,u}(\psi^l) = \begin{cases} \frac{\gamma_o(c_1-c_p)}{1+\gamma_o} & if \;\; Y \geq q_o(\frac{\gamma_o(c_1-c_p)}{1+\gamma_o}) \\ \frac{\gamma_o c_2}{1+\gamma_o} & if \;\; Y < q_o(\frac{\gamma_o c_2}{1+\gamma_o}) \\ q_o^{-1}(Y) & o.w. \end{cases} \tag{3.14}$$

Remarks: The proof is provided in Appendix A. Note that the first case, i.e., $Y \geq q_o(\frac{\gamma_o(c_1-c_p)}{1+\gamma_o})$, refers to a case of supply surplus, i.e., there is more supply than the total demand from traditional and opportunistic energy users, and the second case is tied with a case of supply deficit. Then, it is natural that the real-time price in the first case is lower so as to encourage the consumption of opportunistic energy users, whereas the real-time price in the second case is higher. Note also that, in both cases, when the demand of opportunistic energy users becomes increasingly elastic, the real-time price progressively decreases to the minimum allowable prices, i.e., $c_1 - c_p$ and c_2, respectively. This monotonic behavior of the real-time price with respect to increasing elasticity comes at no surprise, since, as $\gamma_o \to -\infty$, the average demand of opportunistic energy user $q_o(v) \to 0$, i.e., the opportunistic energy users become more and more thrifty. Therefore, the system operator has to offer power at increasingly cheaper prices, up to the lowest possible price. Therefore, when opportunistic demand is elastic, the above pricing scheme could effectively manage opportunistic demand so as to enhance the system reliability.

3.3.3.2 Approximate Solution to the Day-ahead Scheduling Problem

Having established a closed-form real-time pricing policy for both elastic and inelastic cases, a day-ahead schedule can be obtained via the single-stage optimization below:

$$\tilde{\mathcal{P}}^{\mathbf{DA}}_{\mathbf{non-pst}} : \max_{s,u} \mathbf{E}_W \mathbf{E}^u_{D_t} \left[R^l(\psi^l, s, u, \tilde{\vartheta}_{s,u}(\psi^l) \right]. \tag{3.15}$$

where R^l is given by (3.6) and the expectations depend on the exact stochastic models assumed for wind generation and traditional users' energy demand.

Proposition 3.3.3. *The optimal decision of* $\tilde{\mathcal{P}}^{\mathbf{DA}}_{\mathbf{non-pst}}$ *is*

$$\begin{aligned} u^* &= \begin{cases} u_{cap} & \text{if } -1 \le \gamma_t < 0 \\ \frac{\gamma_t}{1+\gamma_t} c_1 & \text{if } \gamma_t < -1, \end{cases} \\ s^* &= \arg\max_s \left\{ (c_2 - c_1)s + \mathbf{E}^v_{W,D} \left[(v - c_2) D_o \right. \right. \\ &\quad \left. \left. - c \mathbf{1}_B (s + W - \alpha_t u^{*\gamma_t} - \varepsilon_t - D_o) \right] \right\}. \end{aligned} \tag{3.16}$$

Remarks: The proof is provided in Appendix A. Based on the results in Proposition 3.3.1, it is easy to see that when opportunistic demand is inelastic, the optimal procurement of base-load generation is given by:

$$s^* = q_o(v_{cap}) + \alpha_t u^{*\gamma_t} - \hat{W} + \mathbf{F}_Z^{-1} \left(1 - c_p/c\right), \tag{3.17}$$

where $\mathbf{F}_Z^{-1}(\cdot)$ denotes the inverse of the of $Z \triangleq \varepsilon_t - \varepsilon_w$. One key observation from (3.17) is that the last term, $\mathbf{F}_Z^{-1}\left(1 - c_p/c\right)$ that equals $s^* - \mathbf{E}_{W,D}[D - W]$ could be regarded as spinning reserve, i.e., the additional generation beside those scheduled to satisfy the expected net demand $\mathbf{E}_{W,D}[D - W]$. Since $\mathbf{F}_Z^{-1}(\cdot)$ is a non-decreasing

function, it is clearly that $\mathbf{F}_Z^{-1}(1 - c_p/c)$ becomes larger as c_p decreases, or $c_2 - c_1$ increases, or the variance of Z increases. In other words, more spinning reserve would be scheduled when the spinning reserve cost is lower, or when the non-spinning reserve cost is higher, or when the uncertainty in net demand is higher. For the elastic case, s^* could be solved using numerical methods, and similar conclusion on reserve can be drawn.

3.4 Multi-timescale Scheduling with Persistent Opportunistic Energy Users

We now study multi-timescale scheduling when the opportunistic energy users are *persistent*. We assume that the opportunistic energy users are persistent across *both* T_2 and T_1 slots, and that the opportunistic energy users that arrived in the day leave the system at the end of the day. Due to the persistent nature of the opportunistic energy users, scheduling decisions in both T_2 and T_1-slots affect the system trajectory and hence scheduling decisions in future slots across both timescales. Thus, the scheduling problem involves hierarchically structured control [45], with the hierarchy defined across timescales. With this insight, we treat the scheduling problem as a *multi-timescale Markov decision process* [46]. where the decisions made in the higher level affects both the state transition dynamics and the decision process at the lower level, while decisions at the lower level affect only the decisions made at the upper level. Further, the MMDP in this study has the following special characteristics: the two timescales *do not overlap*, since the upper level decisions (day-ahead) are made in *non real-time*. Thus, the upper level decisions are made without any direct observation of the effect it has on the lower level system dynamics, and make decisions solely based on stochastic understanding of the lower level process. These properties make the rigorous formulation of the multi-timescale scheduling problem, with persistent users, uniquely challenging. We now describe the problem in detail.

For day-ahead scheduling, the system operator decides the energy supply procurement s_m and the price u_m for the mth T_1-slot in the next day. Recall that the expected amount of wind generation $\hat{W}_m$ is available in day-ahead. We define the system state, ψ_m^u, corresponding to the mth T_1-slot in the next day as $\psi_m^u = \{\hat{W}_m, P_m^u\}$, where P_m^u denotes the number of persistent opportunistic energy users carried over from the $(m-1)$th T_1-slot. During real-time scheduling, the system operator has the knowledge of the wind generation, the demand from traditional energy users and the number of persistent opportunistic energy users carried over from previous T_2-slot. Based on this information, it must decide a real-time price $v_{k,m}$ for the kth T_2-slot in the mth T_1-slot, i.e., the (k,m)th T_2-slot. We define the observable (observable in real time) state of the system in (k,m)th T_2-slot as $\psi_{k,m}^l = \{W_{k,m}, D_{tk,m}, P_{k,m}^l\}$, where $W_{k,m}$ denotes the wind generation in the kth T_2-slot in the mth T_1-slot, $D_{t_{k,m}}$ is the energy demand from traditional energy users in the (k,m)th T_2-slot, conditioned on the day-ahead price, as defined in Section II A, $P_{k,m}^l$ denotes the number of persistent opportunistic energy users carried over from the previous T_2-slot to the (k,m)th slot. Having explicitly defined the system states for the day-ahead scheduling and real-time scheduling, we then introduce the optimality equations. With $\vec{X}_m = [X_{1,m}, \cdots, X_{K,m}]$, we have

$$V_m^u(\psi_m^u) = \max_{s_m,u_m} \Big\{ \mathbf{E}_{\vec{W}_m,\vec{D}_{t_m}} \Big[\max_{\vec{v}_m} \big\{ \mathbf{E}_{\vec{P}_m^l} \sum_{k=1}^{K} \big[R_{k,m}^l(\{\psi^u{}_m, \psi_{k,m}^l\}, s_m, u_m, v_{k,m}) \big] + \mathbf{E}_{P_{m+1}^u = P_{K,m}^l} V_{m+1}^u(\psi_{m+1}^u) \big\} \Big] \Big\} \tag{3.18}$$

As noted earlier, this is an MMDP over a finite horizon, with s_m, u_m being the upper level (slower timescale) decisions and $v_{k,m}$ being the lower level decisions.

To mitigate the complexity of the MMDP problem, we exploit the structural properties of the multi-timescale scheduling problem and recast it as a classic MDP.

Proposition 3.4.4. *With appropriately defined immediate reward R^u_m and action space a^u_m, the two-level scheduling problem can be written as a classic MDP at the slower time-scale, the Bellman equation of which is given by:*

$$V^u_m(\psi^u_m) \;=\; \max_{a^u_m=\{s_m,u_m,\zeta_m\}} \Big\{ R^u_m(\psi^u_m, a^u_m) + \; \mathbf{E}^{\{\psi^u_m,a^u_m\}}_{P^u_{m+1}} V^u_{m+1}(\psi^u_{m+1}) \Big\}, \tag{3.19}$$

where ζ_m:$\{W_{k,m}$,$D_{t_{k,m}}$,$P^l_{k,m}\}\rightarrow v_{k,m}$ is a stationary mapping within the mth T_1-slot, and the various expectations used in the MDP formulation are defined in (3.20).

$$\mathbf{E}^{\{\psi^u_m,a^u_m\}}_{P^u_{m+1}}(.) = \mathbf{E}^{P^l_{1,m}}_{P^l_{2,m}} \mathbf{E}^{P^l_{2,m}}_{P^l_{3,m}} \cdots \mathbf{E}^{P^l_{K-1,m}}_{P^l_{K,m}} \mathbf{E}^{P^l_{K,m}}_{P^u_{m+1}}(.)$$

$$\mathbf{E}^{P^l_{k,m}}_{P^l_{k+1,m}}(.) = \mathbf{E}_{W_{k,m}} \mathbf{E}_{D_{t_{k,m}}} \mathbf{E}_{N_{k,m}} \mathbf{E}^{\{N_{k,m},P^l_{k,m}\}}_{P^l_{k+1,m}}(.)$$

$$\Pi_{v_{k,m}} = P(V \leq v_{k,m})$$

$$\mathbf{E}^{\{N_{k,m},P^l_{k,m}\}}_{P^l_{k+1,m}}(.) = \sum_{P^l_{k+1,m}=0}^{N_{k,m}+P^l_{k,m}} \begin{pmatrix} P^l_{k,m} + N_{k,m} \\ P^l_{k+1,m} \end{pmatrix} (1-\Pi_{v_{k,m}})^{P^l_{k+1,m}} (\Pi_{v_{k,m}})^{(N_{k,m}+P^l_{k,m}-P^l_{k+1,m})}(.) \tag{3.20}$$

We now proceed to discuss the transformation of the two-level scheduling problem from an MMDP to a classic MDP. In the two-level scheduling problem, recall that the lower level decisions are essentially the mapping from the realizations of wind generation, traditional users' energy demand and persistent opportunistic energy users to the real-time price, i.e., $\zeta_{k,m} : \{W_{k,m}, D_{tk,m}, P^l_{k,m}\} \rightarrow v_{k,m}$. Consider a stationary real-time pricing policy within each T_1 slot, i.e., $\zeta_{1,m} = \zeta_{2,m} \ldots \zeta_{K,m}$, and denote this stationary mapping by ζ_m. Thus, $v_{k,m} = \zeta_m(W_{k,m}, D_{t_{k,m}}, P^l_{k,m})$. A key step towards the above results is to view ζ_m as an action at taken day-ahead scheduling, in addition to actions s_m, u_m. With this insight, we can simplify the

MMDP into a classic MDP, as discussed below. Note that in (3.4.4), $V_m^u(\psi_m^u)$ is the expected net reward from slot m until slot M in day-ahead scheduling, and the terminal reward is given by

$$V_M^u(\psi_M^u) = \max_{a_M^u} R_M^u(\psi_M^u, a_M^u). \tag{3.21}$$

Note that the immediate reward corresponding to mth T_1-slot is a function of the realized values of wind generation mean (that is accurately forecast in day-ahead) and the number of persistent opportunistic energy users carried over from previous T_1-slot. We now proceed to explicitly characterize the immediate reward R_m^u, for $m \in \{1, \dots M\}$:

$$R_m^u(\psi_m^u, a_m^u) = V_{k,m}^l(\{\hat{W}_m, P_{k,m}^l\}, a_m^u)|_{k=1}, \tag{3.22}$$

where $P_{1,m}^l = P_m^u$ by definition, and for $k \in \{1, \cdots, K-1\}$

$$V_{k,m}^l(\{\hat{W}_m, P_{k,m}^l\}, a_m^u = \{s_m, u_m, \zeta_m\})$$

$$\begin{aligned} = \ & \mathbf{E}_{W_{k,m}}^{\hat{W}_m} \mathbf{E}_{D_{t_{k,m}}}^{u_m} R_{k,m}^l(\psi_{k,m}^l, a_m^u) \\ & + \mathbf{E}_{P_{k+1,m}^l}^{P_{k,m}^l} V_{k+1,m}^l(\{\hat{W}_m, P_{k+1,m}^l\}, a_m^u), \end{aligned} \tag{3.23}$$

and $V_{K,m}^l$ is given by

$$V_{K,m}^l(\{\hat{W}_m, P_{K,m}^l\}, a_m^u) = \mathbf{E}_{W_{K,m}}^{\hat{W}_m} \mathbf{E}_{D_{t_{K,m}}}^{u_m} R_{K,m}^l(\psi_{K,m}^l, a_m^u), \tag{3.24}$$

where $\mathbf{E}_{W_{K,m}}^{\hat{W}_m}$ and $E_{D_{t_{K,m}}}^{u_m}$ are defined using (1) and (2), respectively. Note that the quantities $R_{k,m}^l$ and $V_{k,m}^l$ can be regarded as the immediate reward and the net reward at the lower level MDP, respectively. The quantity $R_{k,m}^l$ is a function of the realizations of the wind generation, the demand from traditional energy users and

the number of persistent opportunistic energy users carried over from the previous T_2-slot. More specifically, (3.6) can be re-formulated as

$$R^l_{k,m}(\psi^l_{k,m}, a^u_m) = u_m D_{t_{k,n}} + \mathbf{E}_{N_{k,m}} \mathbf{E}^{\{N_{k,m}, P^l_{k,m}, v_{k,m}\}}_{N_{a_{k,m}}} [v_{k,m} D_{o_{k,m}}$$

$$+ \ (-c_p s_m)\mathbf{1}_A + (-\epsilon_{k,m} c_p - (s_m - \epsilon_{k,m})c_1)\mathbf{1}_B$$

$$+ \ (-c_1 s_m + c_2 \epsilon_{k,m})\mathbf{1}_C], \tag{3.25}$$

where $N_{k,m}$ denotes the number of opportunistic energy users arriving at the (k,m)th slot, which is Poisson distributed.

Summarizing, we have rigorously formulated the multi-timescale scheduling problem as an MMDP with special characteristics, and shown that it can be recast explicitly as an MDP with continuous state and action spaces. Using appropriate discretization techniques, we can reformulate it as a classic discrete state and action space MDP that can be solved optimally or near-optimally using various techniques available in the literature [47].

3.5 Numerical Results

We now study, via numerical experiments, the performance of the proposed approach in the multi-timescale scheduling framework, through comparison with existing scheduling schemes in a benchmark smart grid system where *all* the energy users are assumed to exhibit traditional response to the day-ahead prices u with the same price elasticity γ_t. Therefore, the scheduling and pricing decisions $(\bar{s}^*, \bar{u}^*)$ of the benchmark system could easily be obtained from (3.16) by neglecting the opportunistic demand and real-time pricing, i.e., $\bar{s}^* = \bar{\alpha}_t \bar{u}^{*\gamma_t} - \hat{W} + \mathbf{F}_Z^{-1}(1 - c_p/c)$ and $\bar{u}^* = u^*$. The performance metrics are the per unit generation cost, which is defined as the ratio of total generation cost to the total demand served, and system reliability metrics

including loss of load probability (LOLP) and expected energy not served (EENS). Further, we also discuss the impact on the dispatched base-load and fast-start generations.

In this numerical study, we focus only on the key parameters and investigate their impact on the system performance, including the penetration level of wind generation η_w, the penetration level of opportunistic demand η_o and the uncertainty of wind generation. Specifically, as in literatures, the penetration level of wind generation η_w is defined as the ratio of wind generation to the total energy supply over the whole scheduling horizon. Similarly, the penetration level of opportunistic demand η_o is defined as the ratio of opportunistic demand to the total energy demand. Data for other system parameters are described as follows. For simplicity, we assume that the penetration level of opportunistic demand is a constant across T_1 slots.

3.5.1 Simulation Data

3.5.1.1 Stochastic Wind Generation

Wind generation data are collected from [66] and scaled according to the penetration level η_w. The "point-forecast" $\hat{W}_m$ in (3.3) is provided using the Markov chain developed in [66]. For simplicity, the probability distribution illustrated in Fig. 3.2 is used for ε_w. It is easy to see that the forecast error ε_{w_m} has a support $(-3\lambda\hat{W}_m, 3\lambda\hat{W}_m)$, and λ is exactly the MAPE. Therefore, the uncertainty of wind generation could be controlled by varying λ, which is viable by varying the number of states of Markov chain. For day-ahead forecast, λ is usually around 20% [6].

3.5.1.2 Generation Cost

We adopt the approach in [48], which utilizes the heat rates of generators to compute the generation cost. Here, base-load generators of coal type and fast-start generators

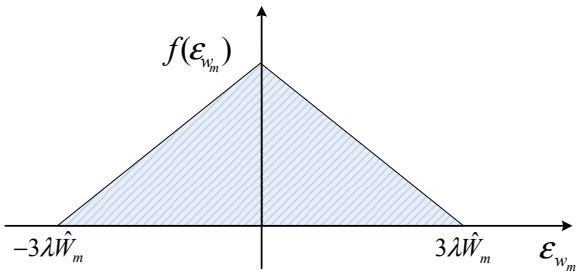


Figure 3.2: PDF of wind generation forecast error in the mth T_1-slot.

of #2 oil type are considered. Specifically, c_1=30.45\$/MWh, c_2=228.51\$/MWh and c_p=15\$/MWh.

3.5.1.3 Uncertain Demand

The hourly demand data in Table. 4 of [49] with peak value 8550 MW is used for both systems. For given total hourly demand D, we set $\mathbf{E}[D_t]$=D for the benchmark system; for the multi-timescale scheduling system, traditional and opportunistic demand are properly scaled by setting $\mathbf{E}[D_t]$=$(1-\eta_o)D$ and $\mathbf{E}[D_o]$=$\eta_o D$, respectively. For ε_t, a zero-mean normal distribution with standard deviation $\tilde{\sigma}_t$ that is 3% of the expected demand $\mathbf{E}[D_{t_m}]$ is used by truncating over $(-3\tilde{\sigma}_t,3\tilde{\sigma}_t)$. Further, we use γ_t=-0.5, γ_o=-2 and 0.05\$/kWh as the price cap for both u and v.

3.5.2 Performance Evaluation and Discussion

We use the case of non-persistent opportunistic demand and the hour with peak demand of 8550 MW as an illustrative example. Note that the metrics are computed via Monte-carlo simulations by choosing wind generation and demand randomly according to their distributions specified earlier.

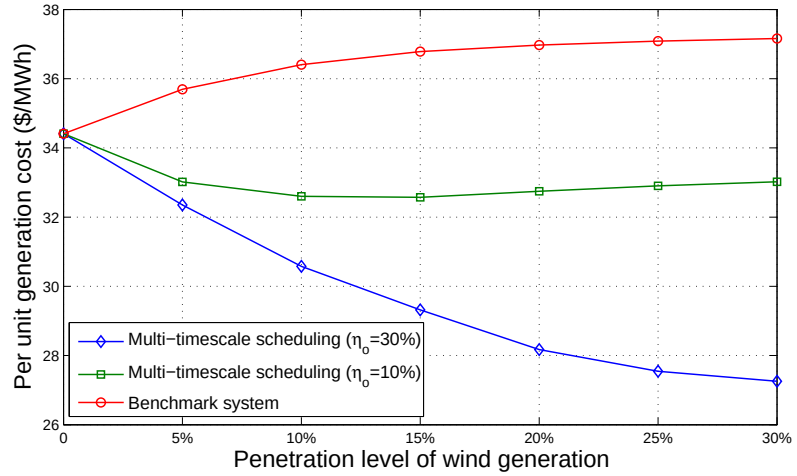


Figure 3.3: Per unit generation cost ($\lambda = 20\%$).

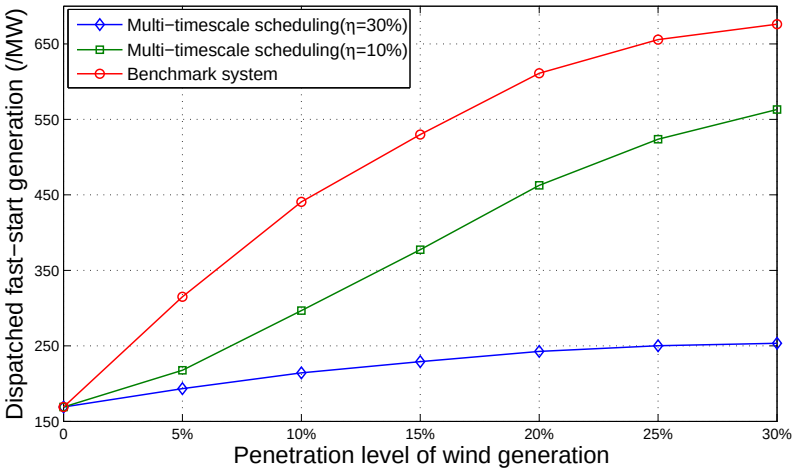


Figure 3.4: Dispatched fast-start generation ($\lambda = 20\%$).

3.5.2.1 Per Unit Generation Cost

In Fig. 3.3, the per unit generation cost is plotted against various η_w. In the multi-timescale scheduling system, the per unit generation cost decreases significantly with η_w, since wind energy is harvested "cost-free"; in contrast, in the benchmark system,

the per unit generation cost is not really reduced. The reason is revealed in Fig. 3.4, i.e., much more fast-start generation is dispatched in the benchmark system.

3.5.2.2 Reliability

In practice, usually, a fixed amount of active reserve is scheduled. One practical rule is the "X+Y" rule [50], in which active reserve is $R{=}X\%\hat{D}{+}Y\%\hat{W}$ for given forecasted demand $\hat{D}$ and wind $\hat{W}$. As discussed earlier, part of s^* is scheduled as spinning reserve of amount $SR{=}s^*{-}\mathbf{E}[D-W]$. Therefore, adhering to a "3+10" rule, we investigate the system reliability by considering $NS{=}R{-}SR$ fast-start generation as active non-spinning reserve, quantified by LOLP, i.e., $\mathbf{E}[\mathbf{1}_{\{D-s^*-W>NS\}}]$, and EENS, i.e., $T_1\mathbf{E}[(D{-}s^*{-}W{-}NS)^+]$. It is observed from the results illustrated in Fig. 3.5 that, for given penetration levels of wind generation, the proposed multi-timescale scheduling approach achieves higher reliability with the same reserve requirement.

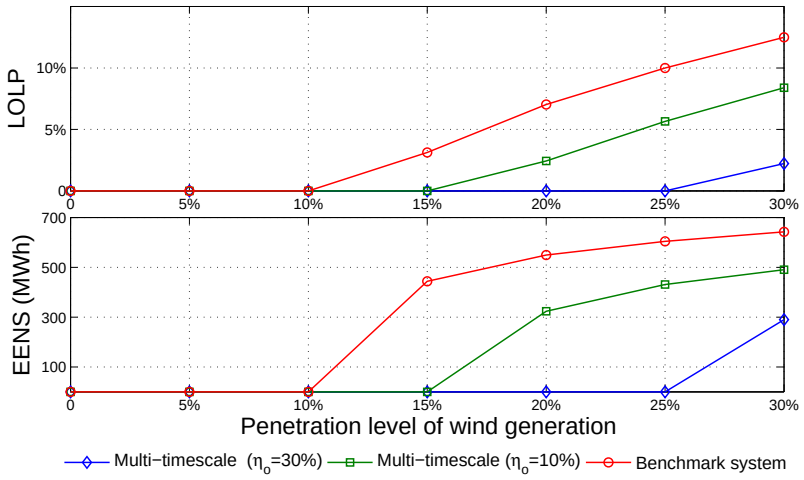


Figure 3.5: System Reliability ($\lambda = 20\%$).

3.5.2.3 Impact of Forecast Accuracy

We control the uncertainty of wind generation due to forecast error by varying λ, and investigate the impact on the fast-start generation that is necessary to maintain the system reliability. It is observed in Fig. 3.6 that the fast-start generation requirement increases with λ in all systems. For multi-timescale scheduling, this is because v keeps fixed and the opportunistic demand could not be reduced any further in the supply deficit case of Proposition 3.3.2). Therefore, it is imperative to develop advanced models and techniques to improve the forecast accuracy.

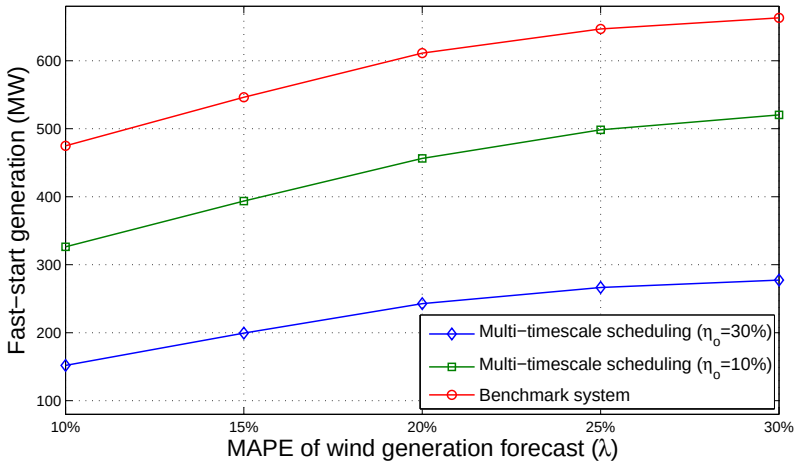


Figure 3.6: Non-spinning reserve requirement ($\eta_w = 20\%$).

Summarizing, the above results suggest that the proposed multi-timescale scheduling schemes have the potential to enhance the system reliability and reduce the cost incurred by wind generation integration. Note also that, the benefit brought by multi-timescale scheduling could be very limited if η_o is low, e.g., when $\eta_o \leq 10\%$, as observed in Fig. 3.3- Fig. 3.6.

3.6 Conclusion

Multi-timescale scheduling and pricing with traditional and opportunistic energy users is investigated to address the challenge of integrating volatile wind generation into smart grids. Specifically, when the opportunistic users are non-persistent, we obtain closed-form solutions to the multi-scale scheduling problem, by assuming two reasonable conditions. When the opportunistic energy users are persistent, we formulate the scheduling problem as an MMDP and discussed its special characteristics. We then show that the scheduling problem can be recast, explicitly, as a classic MDP with continuous state and action spaces, the solution to which can be found via standard MDP solution techniques. The optimal scheduling and pricing decisions are characterized rigorously for both non-persistent model and persistent model. Through numerical experiments, we demonstrate the potential benefit of the proposed multi-timescale scheduling approach, when compared with existing system and schemes.

PART II

SYNCHROPHASOR DATA MINING AND INFORMATION FUSION

Chapter 4

A ROBUST DATA-MINING FRAMEWORK FOR ON-LINE DSA USING ADAPTIVE ENSEMBLE DECISION TREE LEARNING

4.1 Introduction

Dynamic security assessment [51] can provide system operators important information regarding the transient performance of power systems under various possible contingencies. By using the real-time or near real-time measurements collected by phasor measurement units (PMUs), *online* DSA can produce more accurate security classification decisions for the present OC or imminent OCs. However, online DSA still constitutes a challenging task due to the computational complexity incurred by the combinatorial nature of $N-k$ (k=1,2,$\cdots$) contingencies and the massive scale of practical power systems, which makes it intractable to perform power flow analysis and time domain simulations for all contingencies in real-time.

The advent of data mining techniques provides a promising solution to handle these challenges. Cost-effective DSA schemes have been proposed by leveraging the power of data mining tools in classification, with the basic idea as follows. First, a knowledge base is prepared through comprehensive offline studies, in which a number of predicted OCs are used by DSA software packages to create a collection of training cases. Then, the knowledge base is used to train classification models that characterize the decision rules to assess system stability. Finally, the decision rules are used to map the real-time PMU measurements of pre-fault attributes to the security classification decisions of the present OC for online DSA. The data mining tools that have proven effective for DSA include decision trees [52–57], neural networks [58–60] and support vector machines [61–63]. More recently, fuzzy-logic techniques [64] and ensemble learning techniques [65–67] have been utilized to enhance the performance

of these data mining tools in security assessment of power systems. Among various data mining tools, DTs have good interpretability (or transparency) [68], in the sense that the secure operating boundary identified by DTs can be characterized by using only a few critical attributes and corresponding thresholds. As illustrated in Fig. 4.1, a well-trained DT can effectively and quickly produce the security classification decisions for online DSA, since only a few PMU measurements of the critical attributes are needed. The high interpretability of DTs is amenable to operator-assisted preventive and corrective actions against credible contingencies [69]. However, as discussed in [70], there exists an "accuracy versus transparency" trade-off for data mining tools. In order to obtain a more accurate classification model from DTs, one possible approach is to use an ensemble of DTs at the cost of reduced interpretability. Examples of ensembles of DTs for DSA are the multiple optimal DTs [56], random forest [65] and boosting DTs [66].

When applying data-mining-based approaches to online DSA, there are two main issues that can result in inaccurate security classification decisions. First, the realized OCs in online DSA can be dissimilar to those in the initial knowledge base prepared offline, since the predicted OCs might not be accurate and the OCs can change rapidly over time. Second, it is possible that a system topology change may occur during the operating horizon due to the forced outage of generators, transformers and transmission lines. These factors can compromise the performance of the classification model trained offline. To develop a robust data-mining-based online DSA scheme, the initial knowledge base and the classification model have to be updated in a timely manner to track these changed situations. However, there have been limited efforts directed towards handling OC variations and topology changes. In the scheme proposed in [56], when the built DT fails to classify the changed OCs correctly, a new DT is built from scratch or a sub-tree of the DT is replaced by

a newly built corrective DT. Aiming to deal with possible topology changes, references [59, 67] suggest creating an "overall" knowledge base that covers all possible system topologies and choosing the attributes that are independent of topology for data mining.

In this chapter, a robust data-mining-based DSA scheme using adaptive ensemble DT learning is proposed to handle these challenges in a more efficient manner. Specifically, the classification model for DSA is based on boosting multiple *unpruned small-height* DTs[1]. Generally, the height of a DT is the maximal number of tests that is needed for the DT to classify a case. For the sake of brevity, small DTs are referred to as small DTs throughout. In offline training, the small DTs and their voting weights are sequentially identified in a "gradient-descent" manner to minimize the misclassification cost. The small DTs, together with their voting weights, are then periodically updated throughout the operating horizon by using new training cases that are created to account for any change in OC or network topology. Different from existing DT-based DSA schemes, the training cases are assigned different data weights by each small DT; and higher data weights are assigned to a new training case if it is misclassified by the small DTs. The aforementioned techniques are utilized to minimize the misclassification cost as new training cases are added to the knowledge base, so that the classification model could smoothly track the changes in OCs or system topology.

The rest of this chapter is organized as follows. A brief introduction to DTs and their application to DSA are given in Section 4.2. The proposed scheme is discussed in detail in Section 4.3. An illustrative example by using the IEEE 39-bus test system is presented in Section 4.4. The proposed scheme is applied to the WECC system in Section 4.5. Finally, conclusions are provided in Section 4.6.

[1]For the sake of abbreviation, small-height DTs are referred to as small DTs in the remainder of this disseration

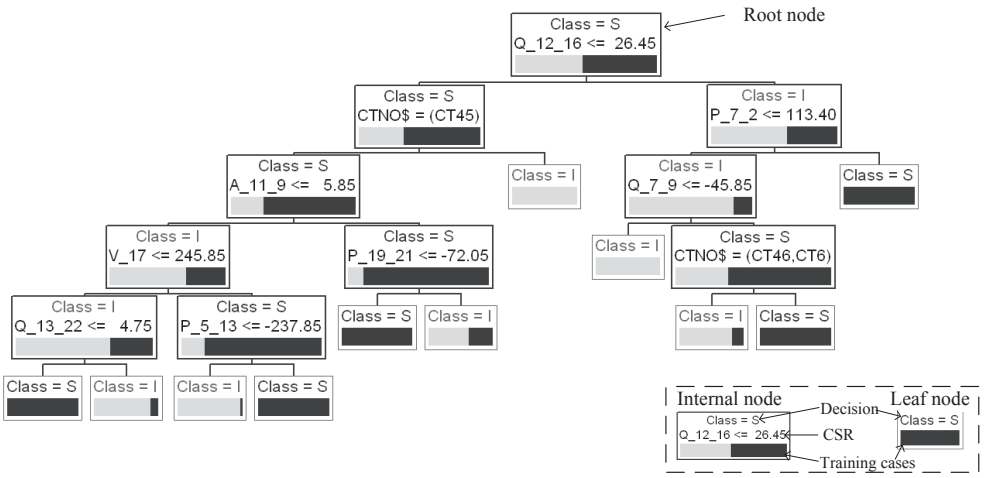


Figure 4.1: A fully-grown DT of height 5 for the WECC system using an initial knowledge base consisting of 481 OCs and three critical contingencies.

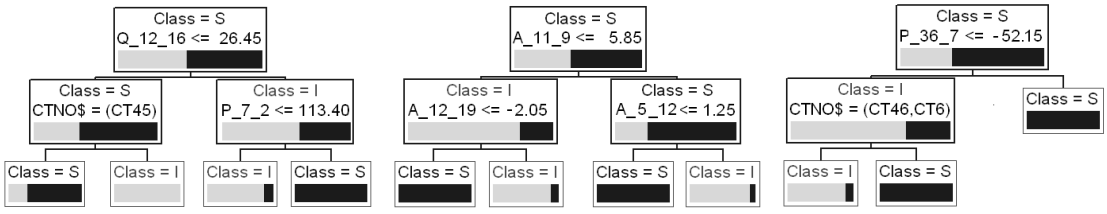
The data-mining framework for DSA was originally developed in [52], in which DTs were introduced to perform DSA for power systems. A DT, as illustrated in Fig. 4.1, is a tree-structured predictive model that maps the measurements of an attribute vector $\mathbf{x}$ to a predicted value $\hat{y}$. When DTs are used for online DSA, the attribute vector can consist of various PMU-measured variables and other system information, and the binary decision given by DTs represents the security classification decision of an OC for a critical contingency (e.g., $\hat{y}$=+1 represents the insecure case, and $\hat{y}$=−1 for the secure case). Usually, bus voltage phase angles, bus voltage magnitudes and branch power/current flows that are directly measured by PMUs are used as numerical attributes. Fig. 4.1 illustrates the numerical and categorical attributes used in a trained DT, in which an attribute with initial "V" stand for a bus voltage magnitude, the attributes with initials "P", "Q", and "A" stand for an active power flow, a reactive power flow, and a voltage phase angle difference between two buses, respectively (the bus numbers in attribute names are different from their real ones), "CTNO$" stands for the index of contingency.

In a DT, each non-leaf node tests the measurement of an attribute and decides which child node to drop the measurements into, and each leaf node corresponds to a predicted value. As shown in Fig. 4.1, in a DT for DSA, the predictive value of each leaf node is either "S" or "I", in which "S" stands for secure cases and "I" for insecure cases. Fig. 4.1 also illustrates the training cases that fall into each node, by using dark bars for secure cases and bright bars for insecure cases. The number of non-leaf nodes along the longest downward path from the root node to a leaf node is defined as the *height* of a DT. Given a collection of training cases $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$, the objective of DT induction is to find a DT that can fit the training data and accurately predict the decisions for new cases. State-of-the-art DT induction algorithms are often based on greedy search. For example, in the classification and regression tree (CART) algorithm [71], the DT grows by recursively splitting the training set and choosing the critical attributes (numerical or categorical) and critical splitting rules (CSR) with the least splitting costs until some predefined stopping criterion (e.g., the size of tree or the number of training cases in a leaf node) is satisfied. In general, a fully-grown DT that accurately classifies the training cases might misclassify new cases outside the knowledge base. This feature of fully-grown DTs is usually referred to as "overfitting" [68]. In order to avoid overfitting, DTs are usually pruned by collapsing unnecessary sub-trees into leaf nodes. As illustrated in Fig. 4.1, in a pruned DT, some leaf nodes do not have pure training cases, which is a result of either tree pruning or early termination of tree growing [68]. By removing the nodes that may have grown based on noisy or erroneous data, the pruned DT is more resistant to overfitting than a fully-grown DT without pruning, and thus can give more accurate security decisions

A major advancement in DT-based DSA schemes was made in [57], in which the authors proposed to build a single DT to handle multiple contingencies, by using the index of contingencies as a categorical attribute of the DT. It is worth noting that a

DT built by using such an approach can give the security classification decisions of an OC concurrently for all the critical contingencies in the knowledge base, which is more efficient and can identify the critical attributes that are independent of contingencies. For example, the DT in Fig. 4.1, using CTNO$ as a categorical attribute, can give security classification decisions of an OC for three critical contingencies, i.e., CT6, CT45 and CT46, at the same time, and the critical attributes $Q_{12,16}$, $P_{7,2}$, $Q_{7,9}$, $A_{11,9}$, $A_{12,19}$, $A_{5,12}$ and $P_{36,7}$ can give security classification decisions independent of contingence type for some cases.

4.2.1 Small DTs



(a) Small DT h_1 (b) Small DT h_2 (c) Small DT h_3

Figure 4.2: The first three small DTs (J=2) for the WECC system, the voting weights of which are 4.38, 3.04 and 0.93, respectively.

A small DT with tree height J is obtained by stopping the splitting of any leaf node if the downward path from the root node to that leaf node has exactly J non-leaf nodes. According to [72], a small DT is much less prone to overfitting compared to a fully-grown DT; therefore, the small DTs used in the proposed scheme are built without pruning. Examples of small DTs are given in Fig. 4.2 with J=2. It can be seen that the non-leaf nodes of h_1 are exactly the same as the corresponding nodes of the DT in Fig. 4.1. It is worth noting that the optimal choice of J is highly dependent on the knowledge base, and should be decided based on a bias-variance analysis [68], which will be discussed in the case study of Section IV. Note also that different from [68], the tree height, instead of the number of nodes, is used as the

metric to quantify the tree size. The reason, which will be soon apparent, is to restrict the number of nodes that will be revised when updating DTs to a value less than J.

4.2.2 Ensemble of DTs

In ensemble-DT-based DSA schemes, the security classification decision of an OC vector $\mathbf{x}$, denoted by $H_L(\mathbf{x})$, is made based on the voting of multiple DTs. For an ensemble of DTs $h_l(l = 1,2,\cdots,L)$, there are two approaches to DSA classification: deterministic and probabilistic. For the deterministic approach, the security classification decision is given by:

$$H_L(\mathbf{x}) = \begin{cases} +1, & \text{if } \sum_{l=1}^{L} a_l h_l(\mathbf{x}) \geq 0 \\ -1, & \text{o.w.} \end{cases} \tag{4.1}$$

where a_l $(l = 1,2,\cdots,L)$ are the voting weights of DTs. To obtain probabilistic classification decisions, the "logistic correction" technique [73] can be applied. Then, the probability of an "Insecure" classification decision is given by:

$$\Pr(H_L(\mathbf{x}) = +1|\mathbf{x}) = \frac{1}{1 + \exp(-\sum_{l=1}^{L} a_l h_l(\mathbf{x}))} \tag{4.2}$$

In this chapter, deterministic classification decision is used to calculate the misclassification rate for case studies.

The existing methods for ensemble DT learning include bagging, random subspace method, boosting and random forest. Reference [74] compares these methods, and finds that boosting and random forest achieve significantly better performance than the others. In previous work by the authors [66], an algorithm for boosting DTs is developed in the context of avoiding overfitting to noisy training data. In this chapter, the boosting algorithm is employed in online DSA to deal with OC varia-

tions and possible topology changes. The algorithm for building the small DTs and calculating the voting weights will be discussed in Section III.A.

4.2.3 Updating DTs

One existing approach for updating a DT without rebuilding it from scratch is the efficient tree restructuring algorithm [75], with the main idea summarized as follows. When incorporating a new case, the DT remains unchanged if the new case is classified correctly; otherwise, the non-leaf nodes along the path which the new case passes are revised in a top-down manner. Specifically, for each non-leaf node to be revised, a new test is first identified by using the new case as well as the existing cases that fall into the non-leaf node. If different from the original test, the newly identified test is then installed at the non-leaf node, followed by tree restructuring operations recursively applied on the sub-tree corresponding to that non-leaf node (there are six slightly different restructuring operations for various structures of the sub-tree, which are not discussed here). The motivation for these restructuring operations is that the original test at the non-leaf node is highly likely to be the optimal tests for the two child nodes after restructuring, which is usually the case when categorical attributes are used by the test [75]; in this scenario, the two child nodes are exempted from further update.

4.3 Proposed Scheme for Online DSA

The proposed scheme for online DSA, as illustrated in Fig. 4.3, consists of three steps, with the details described below.

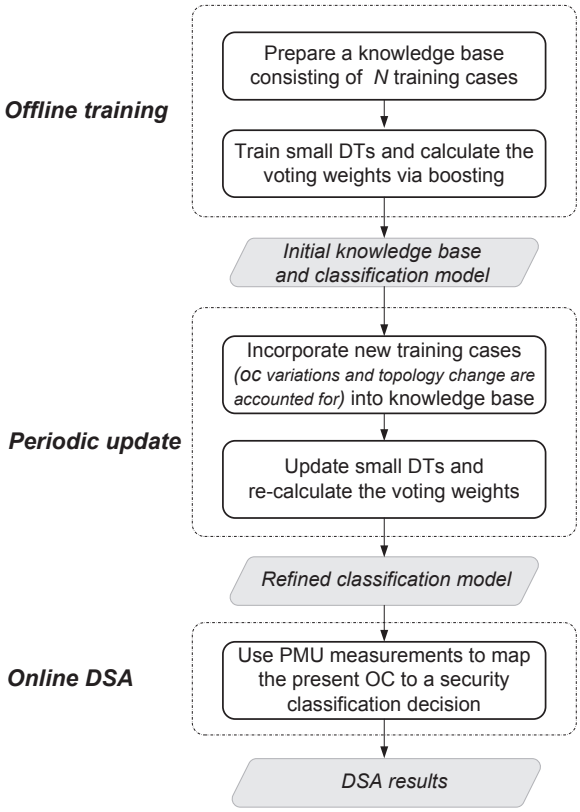


Figure 4.3: Proposed scheme for online DSA using adaptive ensemble DT learning.

4.3.1 Offline Training

4.3.1.1 Initial Knowledge Base Preparation

First, N_{OC} predicted OCs are generated day ahead for each period of the future operating horizon (e.g., the next 24 hours) based on day-ahead load forecast and generation schedules; each period may span several hours, and can be divided according to the hours of peak load, shoulder load and off-peak load. Then, for each of the N_{OC} day-ahead predicted OCs, detailed power flow analysis and time-domain simulations are performed for K critical contingencies that are selected by the system operator or based on prior experience. It is worth noting that the key focus here is on dealing with OC variations and possible topology changes, and thus the selection or screen-

ing of critical contingencies is beyond the scope of this study. By using specified dynamic security criteria (e.g., transient stability, damping performance, transient voltage drop/rise, transient frequency, relay margin), the day-ahead predicted OCs are labeled as "Secure" or "Insecure" for each critical contingency.

As a result, an initial knowledge base that consists of $N = N_{OC} \times K$ training cases is obtained, in which each case is represented by a vector $\{x_1, \cdots, x_P, y\}$, where x_1 is the index of a critical contingency, $\{x_2, \cdots, x_P\}$ are the values of numerical attributes obtained from power flow analysis of an OC, and y is the transient security classification decision of the OC for the critical contingency x_1. Based on the previous studies [55–57], the following PMU-measured variables are selected as numerical attributes:

- Branch active power flows $\{P_{ij};\ i \in \mathcal{B} \text{ or } j \in \mathcal{B}\}$
- Branch reactive power flows $\{Q_{ij};\ i \in \mathcal{B} \text{ or } j \in \mathcal{B}\}$
- Branch current flows (magnitude) $\{I_{ij};\ i \in \mathcal{B} \text{ or } j \in \mathcal{B}\}$
- Bus voltage magnitudes $\{V_i;\ i \in \mathcal{B}\}$
- Bus voltage phase angle differences $\{A_{ij} \triangleq A_i - A_j;\ i, j \in \mathcal{B} \text{ and } i > j\}$,

where $\mathcal{B}$ denotes the set of PMU buses in the system. It is worth noting that only raw measurements reported by PMUs are used as the numerical attributes in this work; more generally, the variables computed using other system information may also be used, e.g., the voltage at the bus connected to a PMU bus when the branch impedance is constant [55].

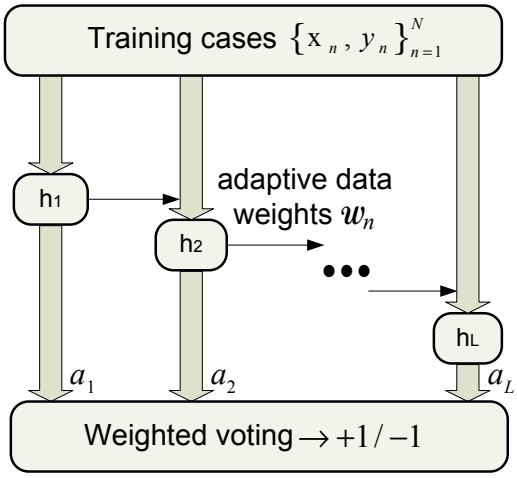


Figure 4.4: Boosting small DTs

4.3.1.2 Boosting Small DTs

The basic algorithmic flowchart of boosting small DTs is illustrated in Fig. 4.4. For convenience, define $\mathcal{H}_J$ as the class of small DTs with height J, define F_L as the score of the weighted voting of the ensemble of small DTs, i.e., $F_L(\mathbf{x})=\sum_{l=1}^{L} a_l h_l(\mathbf{x})$, and define $C_N(F_L)$ as the cost function of F_L on the N training cases, given by:

$$C_N(F_L) = \frac{1}{N}\sum_{n=1}^{N} \log_2(1 + \mathrm{e}^{-y_n F_L(\mathbf{x_n})}). \tag{4.3}$$

It is observed from (4.1) and (4.3) that $C_N(F_L)$ lies strictly above the misclassification error rate of H_L. Then, a primary objective of boosting is to minimize $C_N(F_L)$, by identifying the small DTs $h_l \in \mathcal{H}_J$ and their voting weights $a_l \in \mathcal{R}^+$. An analytical formulation is provided below:

$$\mathcal{P}_F : \min_{\substack{h_1,\cdots,h_L \in \mathcal{H}_J \\ a_1,\cdots,a_L \in \mathcal{R}^+}} C_N(F_L). \tag{4.4}$$

The convexity and the differentiability of $C_N(F_L)$ with regard to F_L make it possible to solve $\mathcal{P}_F$ in (4.4) by using a line search strategy [76], the details of which are summarized as follows. A small DT h_l is chosen to be the "gradient" of C_N at F_{l-1} projected onto $\mathcal{H}_J$, and the voting weight a_l is computed as the "step size" that minimizes $C_N(F_{l-1}+a_l h_l)$. Then, the small DT h_l is added to F_{l-1} to obtain F_l=F_{l-1}+$a_l h_l$. The above steps are iterated, for l=1,2,$\cdots$,L, by using F_0 as a zero function. More specifically, it is shown in [66] that the small DT h_l can be obtained by solving the following problem:

$$\mathcal{P}_h^{(l)}: \min_{h_l \in \mathcal{H}_J} \frac{1}{N} \sum_{n=1}^{N} w_n^{(l)} \mathbf{1}_{\{y_n \neq h_l(\mathbf{x}_n)\}}, \tag{4.5}$$

where $w_n^{(l)} \triangleq (1+\mathrm{e}^{y_n F_{l-1}(\mathbf{x}_n)})^{-1}$ is the positive data weight of the training case $\{\mathbf{x}_n, y_n\}$, and $\mathbf{1}_{\{y_n \neq h_l(\mathbf{x}_n)\}}$ takes value 0 if the training case $\{\mathbf{x}_n, y_n\}$ is correctly classified by the small DT h_l (otherwise, it takes value 1). By definition of $w_n^{(l)}$, it is easy to observe that the data weights are assigned *adaptively* by small DTs, in the sense that if the training case $\{\mathbf{x}_n, y_n\}$ is misclassified by the small DT h_l, then $w_n^{(l+1)} > w_n^{(l)}$, i.e., the training case has a higher data weight in the next round of the boosting process. It is worth noting that highly skewed training data (e.g., the case in [65]) can be handled by scaling up the weights of under-represented cases, such that $\sum_{y=+1} w_n^{(l)} = \sum_{y=-1} w_n^{(l)}$. As suggested in (4.5), the objective of $\mathcal{P}_h^{(l)}$ is to determine the small DT that has the least misclassification error rate on the weighted training data. Thus, the small DT h_l can be obtained by employing the standard CART algorithm [71] subject to the tree height J, and by using misclassification error rate as the splitting cost when building the DT. Then, its positive voting weight is obtained by solving the following problem:

$$\mathcal{P}_a^{(l)}: \min_{a \in \mathcal{R}^+} G_N^{(l)}(a), \tag{4.6}$$

where $G_N^{(l)}(a) \triangleq C_N(F_{l-1} + ah_l)$. Under the condition that h_l is a "descent direction" of $C_N(F_{l-1})$, it is easy to verify that $G_N^{(l)'}(0)<0$ and $G_N^{(l)''}(a)>0$ holds for any $a \in \mathcal{R}^+$. Therefore, $G_N^{(l)}(a)$ has a unique minimum in $\mathcal{R}^+$ that can be found using standard numerical solution methods (e.g., Newton's method).

4.3.2 Periodic Updates

4.3.2.1 New Training Case Creation

In the initial knowledge base prepared offline, the predicted OCs generated using day-ahead forecast may not reflect the actual system conditions, which is very likely to be the case for power systems with high penetration of variable renewable generation and distributed generation. Therefore, as the operating horizon is approached and the data available to system operators is updated, it will be necessary to utilize short-term forecast and schedules to generate newly changed OCs and add them to the knowledge base on a slot-by-slot basis (one slot may span several minutes depending on the processing speed [55]). Further, in case of a topology change, the post-disturbance OCs should also be incorporated into the knowledge base. After power flow analysis of these newly changed OCs, new training cases are generated as described in Section III-A-1. It is worth noting that during the operating horizon, it is also likely that the knowledge base may need to be updated by incorporating new contingencies of interest. The solution to this problem has been discussed in [66]. In this work, the critical contingency list is assumed to remain unchanged during the operating horizon.

4.3.2.2 Updating the Classification Model

Given the newly created training cases, the classification model is updated by using one new case at a time. Specifically, for the k-th new training case $\{\mathbf{x}_{N+k}, y_{N+k}\}$, the classification model is updated by incorporating $\{\mathbf{x}_{N+k}, y_{N+k}\}$ with a data weight

$w_{N+k}^{(l)}=(1+e^{y_{N+k}F_{l-1}(\mathbf{x}_{N+k})})^{-1}$ into the small DT h_l and recalculating the voting weight a_l, iteratively for $l = 1,2,\cdots,L$.

A key step for incorporating a new training case into a small DT is to adopt the method described in Section II-C. Since misclassification error rate is used as the metric of splitting cost, as suggested in (4.5), it is easy to observe that there exists a even simpler solution for updating the small DTs. Specifically, a small DT remains unchanged if the new case is correctly classified; otherwise, only the subtree corresponding to the first non-leaf node that has a different decision for the new case is subject to update. It is worth noting that, since the tree height is J, the total number of non-leaf nodes to be revised is at most J. After the small DT h_l is updated, its voting weight a_l is recalculated by minimizing $G_{N+k}^{(l)}(a)$.

The process of updating the classification model is summarized in Algorithm 1. It is useful to note that when the k-th new training case is used to update the small DTs, the data weights of the previous $N+k-1$ training cases calculated in Step 4 of Algorithm 1 are different from the data weights that were used in building or updating the small DTs in the past rounds. Therefore, unlike the case in offline training, it is possible that the updated small DT h_l is not a "descent direction" of C_{N+k} at F_{l-1} any more. In order to detect and handle this situation, an extra step is used in Algorithm 1. Specifically, if $\sum_{n=1}^{N+k} w_n^{(l)} y_n h_l(\mathbf{x}_n)<0$, then $-h_l$ is a "descent direction" and used for weighted voting.

4.3.3 Online DSA

In real-time, when the synchronized PMU measurements are received, the pre-fault values of the numerical attributes are retrieved and combined with the indices of all critical contingencies to create K unlabeled cases, which will be used by the classification model to give security classification decisions of the present OC for the

Algorithm 1 Periodic updates using a new training case

Input: A new training case $\{\mathbf{x}_{N+k}, y_{N+k}\}$.
Initialization: $F_0 = \mathbf{0}$.
for $l = 1$ to L **do**
  Recalculate the data weights of $\{\mathbf{x}_n, y_n\}_{n=1}^{N+k-1}$.
  Incorporate $\{\mathbf{x}_{N+k}, y_{N+k}\}$ with weight $w_{N+k}^{(l)}$ into h_l.
  Calculate $\varepsilon = \frac{1}{N+k} \sum_{n=1}^{N+k} w_n^{(l)} y_n h_l(\mathbf{x}_n)$.
  if $\varepsilon < 0$ **then**
    $h_l \leftarrow -h_l$.
  end if
  Recalculate a_l by minimizing $G_{N+k}^{(l)}(a)$.
  $F_l \leftarrow F_{l-1} + a_l h_l$.
end for

K critical contingencies. Specifically, when an unlabeled case is processed by the classification model, each of the small DTs uses the values of the attribute vector and its CSRs to produce a binary decision. Finally, the binary decisions of all small DTs are collected and used to give the security classification decisions of the present OC, according to (4.1). It is worth noting that distributed processing technologies [77] can be leveraged to speed up online DSA. Specifically, the K unlabeled cases can be classified separately by using K duplicates of the classification model, and in each classification model, all small DTs can process the attribute vector of an unlabeled case in a parallel manner.

From the above development, it can be seen that the proposed scheme illustrated in Fig. 4.3 is derived from those in previous work [55–57], with the following major modifications. **a**) The classification model is obtained via boosting multiple small unpruned DTs instead of a single fully-grown DT after pruning. It is suggested that boosting algorithms can lead to better model fitting and the produced classification model is quite resistant to overfitting [72]. Thus, boosting small DTs has great potential to deliver better performance in terms of classification accuracy. **b**) Unequal data weights are assigned to the training cases adaptively by small DTs. In

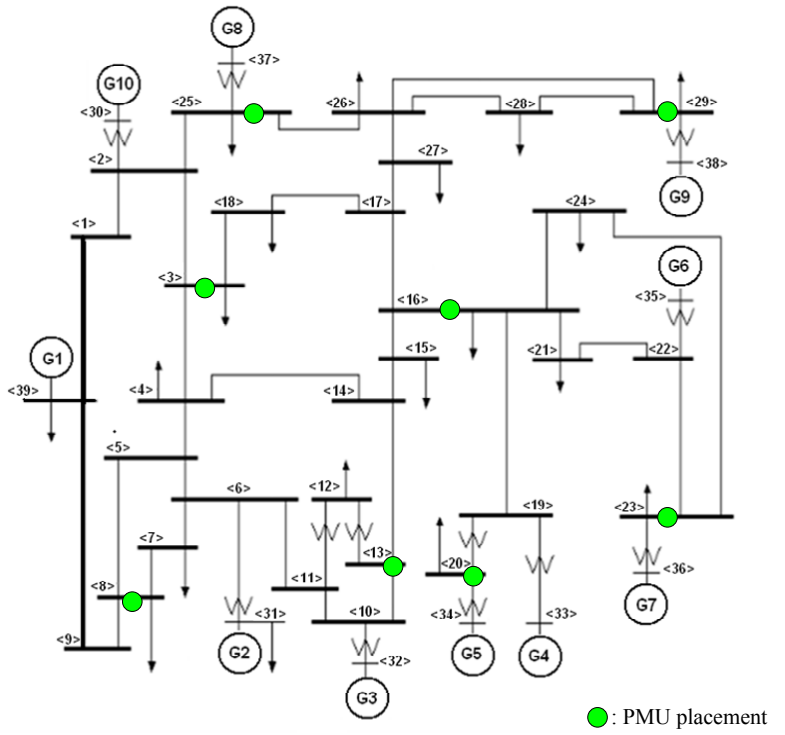


Figure 4.5: The IEEE 39-bus system with 8 PMUs

periodic updates, misclassified new training cases can have higher data weights than those classified correctly. This will speed up adapting the small DTs to newly changed OCs. **c**) The small DTs are gracefully updated by incorporating new cases one at a time, whereas rebuilding DTs is used in [55–57]. **d**) The DT and the knowledge base are updated only when the new cases are misclassified in [55–57]; whereas all new training cases are incorporated into the knowledge base in the proposed scheme.

4.4 An Illustrative Example

The IEEE 39-bus test system [78] is used as an illustrative small system. As illustrated in Fig. 4.5, 8 PMUs are installed in the system, according to the placement design provided in [79]. In what follows, the main steps of the proposed approach, including attribute selection, knowledge base preparation and ensemble small DT learning,

will be demonstrated by using the IEEE 39-bus test system. Finally, the results of robustness test on changed OCs will be presented.

4.4.1 Knowledge Base

4.4.1.1 Attribute Selection

Based on the PMU placement and system topology in Fig. 4.5, 111 numerical attributes are selected according to the rules described in Section III-A, including:

- 8 bus voltage magnitudes at the 8 PMU buses;
- 75 branch active/reactive power flows and current flows, which take any of the 8 PMU buses as either a from-bus or a to-bus of the branch;
- 28 bus voltage phase angle differences, which are computed from the $\frac{8(8-1)}{2}$ pairs of phase angles.

4.4.1.2 OC Generation and Contingencies

The OC specified in [78] is used as the base OC. To enrich the knowledge base, more OCs are generated by randomly changing the bus loads (both active and reactive) within 90% to 110% of their original values in the base OC. For each generated OC, limit checking is carried out by using the power flow and short circuit analysis tool (PSAT) [80], so that any generated OC with pre-contingency overloading or violation of voltage magnitude/angle limits is not included in the knowledge base. Further, transient stability assessment is carried out for the 30 $N-2$ contingencies listed in Table. II of [81]. These $N-2$ contingencies, which can lead to stressed system conditions, are identified by exhaustive search among all possible $N-2$ contingencies.

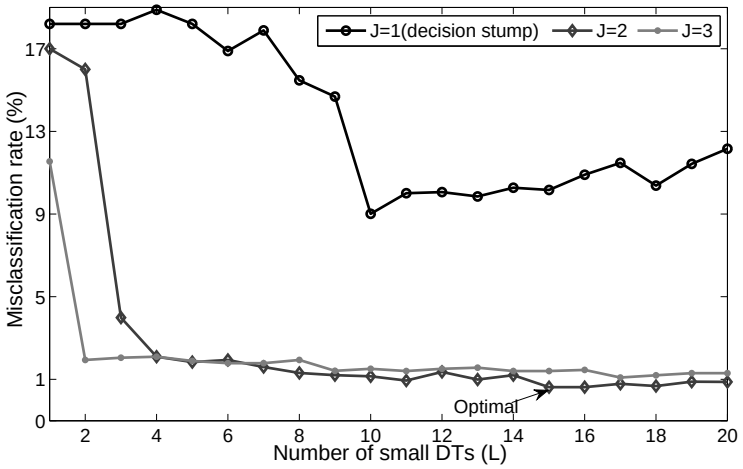


Figure 4.6: Ensemble small DT learning with different tree heights for the IEEE 39-bus test system.

4.4.1.3 Transient Stability Assessment Tool and Criteria

The transient security assessment tool (TSAT) [80] is used to assess the transient performance of the generated OCs. The time-domain simulation is executed for 10 seconds with a step size of 0.5 cycle. The power angle-based stability margin is used as the transient stability index (TSI), defined as:

$$\eta = \frac{360 - \delta_{max}}{360 + \delta_{max}} \times 100, \quad -100 < \eta < 100, \tag{4.7}$$

where δ_{max} is the maximum angle separation of any two generators in the system at the same time in the post-fault response. In case of islanding, the above value is evaluated for each island and the smallest value is taken as the TSI. During the simulation time, whenever the margin η turns out to be negative, i.e., the rotor angle difference of any two generators exceeds 360 degree, the case is labeled as transiently insecure.

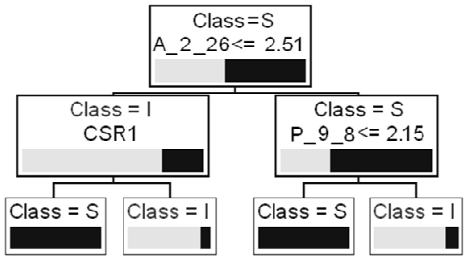
4.4.2 Offline Training

4.4.2.1 Choice of J and L

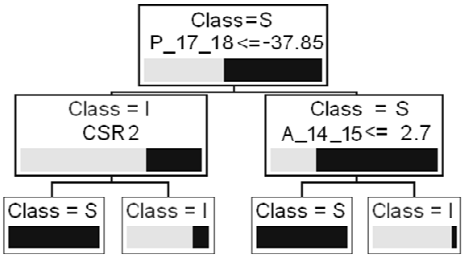
V-fold cross validation (V=10) is carried out to determine the optimal tree height J and the optimal number of small DTs L. Specifically, the training cases in the initial knowledge base are randomly partitioned into V subsets of equal size. For given fixed J and L, a classification model is trained by using $V-1$ subsets, and tested using the other subset. The training process is then repeated V times in total, with each of the V subsets used exactly once as the test data. Finally, the misclassification error rate obtained by V-fold cross validation is calculated by averaging over the V classification models. The results of the above procedure for different tree heights (J=1, 2, 3) are illustrated in Fig. 4.6. It can be seen that as L increases, the misclassification error rate of each classification model decreases and reaches a plateau at some L. Then, when L grows larger, each classification model incurs a larger variance and hence a higher misclassification error rate. On the other hand, a larger tree height J implies a larger variance of classification model [68], which is also observed in Fig. 4.6. Based on these observations, J=2 is chosen, and L=15 at which the misclassification error rate drops below 1% and reaches a plateau is selected.

4.4.2.2 Ensemble Small DT Learning

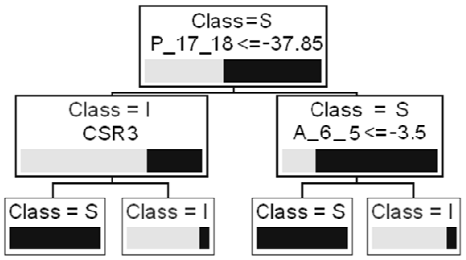
When the optimal tree height J and the optimal number of small DTs L are determined, the algorithm described in Section III-A-2 is used to build the ensemble of small DTs. Specifically, for $l = 1, 2, \cdots, L$, the data weights $w_n^{(l)}$ are first computed according to (5). Then, the training cases together with their data weights are used by the CART algorithm to build a small DT h_l with height J, by using weighted misclassification rate as the cost function, as shown in (5). Note that *each small*



(a) Trained small DT h_1 (CSR1 represents the critical splitting rule: CTNO$ = (CT1, CT2, CT3, CT7, CT11, CT12, CT15, CT16, CT17))



(b) Small DT h_1 updated with changed OCs (CSR2 represents: CTNO$ = (CT1, CT2, CT3, CT4, CT7, CT8, CT12, CT15, CT16, CT19))



(c) Small DT h_1 rebuilt with changed OCs (CSR3 represents: CTNO$ = (CT1, CT2, CT3, CT4, CT7, CT9, CT11, CT15, CT17))

Figure 4.7: The first small DT h_1 (J=2) for the IEEE 39-bus test system.

DT gives security classification decisions for all critical contingencies. Further, the voting weight of h_l is calculated by numerically solving (6). Then, the ensemble of small DTs are obtained. It is worth noting that, different from the V-fold cross validation procedure, the entire training set (not a subset) is used by each small DT of the ensemble.

4.4.3 Robustness Testing

4.4.3.1 Changed OCs

In the IEEE 39-bus test system, generator **G1**, together with transmission lines (39, 9) and (39, 1), represents the equivalent to the external system of the New England area [78]. It is now assumed that the capacity of **G1** reduces from 1100 MW to

900 MW, which could be the result of either the loss of a transmission corridor or a generator tripping outside the New England area. Therefore, the OCs will change due to generation rescheduling. By setting the capacity of **G1** to 900 MW, changed OCs are generated by rescheduling generation and re-solving power flows for each OC in the initial knowledge base. These changed OCs will be utilized to test the robustness of the proposed approach.

4.4.3.2 Robustness Testing Results

First, 200 OCs are generated to create the initial knowledge base consisting of 6000 (200 OCs $\times$ 30 contingencies) training cases. Accordingly, another 200 changed OCs are generated, in which 100 OCs are used to update the small DTs and the other 100 OCs are used for robustness testing. In the proposed approach, Algorithm 1 is applied to update each of the 15 small DTs by using the 3000 (100 OCs $\times$ 30 contingencies) new cases. To illustrate the change of small DTs, the first small DT h_1 is used as an example. Specifically, h_1 obtained in offline training and updated with the 100 changed OCs by using the proposed approach are illustrated in Fig.4.7(a) and Fig.4.7(b), respectively. It is observed that due to the changed OCs and generation rescheduling, the critical attribute in the root node of h_1 changes from the voltage phase angle difference between bus 2 and bus 26, A_2_26, to the active power flow between bus 17 and bus 18, P_17_18. The CSRs of the non-root nodes change accordingly, as a result of the recursive procedure of the CART algorithm. The small DT h_1 rebuilt with the 100 changed OCs is illustrated in Fig.4.7(c), which has the same CSR at the root node as the small DT updated by using the proposed approach. Since the small DTs h_1 obtained by updating and rebuilding are different at non-root nodes, the other small DTs, h_2 to h_{15} are also different. This is because the ensemble DT learning algorithm sequentially updates/builds the small DTs, in which each small DT depends on the previous small DTs.

Table 4.1: Misclassification Error Rate of Robustness Testing

	secure cases	insecure cases	overall
Proposed	0.68%	0.36 %	0.55%
Small DTs (rebuilding)	0.59%	0.38%	0.54%
Small DTs (no updating)	10.68%	6.85%	9.57%

The proposed approach is compared with two benchmark approaches: 1) small DTs rebuilt by using the 100 changed OCs together with the initial 200 OCs, 2) small DTs without updating. The test results of the three approaches are presented in Table. 4.1. It can be seen that the proposed approach achieves comparable performance to the benchmark approach by rebuilding small DTs. The test results also suggest that when OCs change, the small DTs have to be updated in order to track the variation of OCs.

4.5 Application to the WECC System

The test power system used in this case study is part of the Western Electricity Coordinating Council (WECC) system. It consists of over 600 buses (of which 33 are PMU buses), 700 transmission lines and 100 generators.

4.5.1 Knowledge Base

4.5.1.1 OC Generation

The OCs used in the case study are generated by using real-life data of power flows, bus loads and generator power outputs that were recorded every 15 minutes during a 2008 summer peak day. The overall load profile is illustrated in Fig. 4.8. Based on the variations of the aggregate load, each period for offline training is chosen to span 8 hours, and the peak load period 12:00 Hrs-20:00 Hrs is investigated in this case study. Basically, there are three sets of generated OCs used in this case study: day-ahead predicted OCs, short-term predicted OCs and realized OCs. The day-ahead

predicted OCs are used to create the initial knowledge base, the short-term predicted OCs are used to create the new training cases to update the knowledge base and the classification model, and the realized OCs are used for testing purposes only.

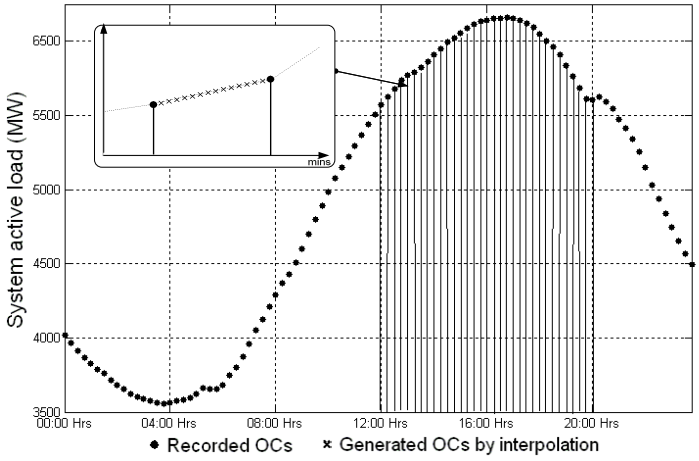


Figure 4.8: Aggregate load of recorded OCs and generated OCs by interpolation.

In what follows, the procedure for generating the three OC sets is discussed in detail. The realized OCs include the 33 recorded OCs and another 448 OCs that are generated by interpolation, as illustrated in Fig. 4.8. Specifically, following the method in [57], both the active and reactive load of each load bus for every minute of the investigated period are obtained by linear interpolation based on the two closest recorded OCs, and the generator power outputs are adjusted as needed to ensure valid OCs. To enrich the initial knowledge base, a day-ahead predicted OC is obtained by randomly changing the bus loads within 90% to 110% of the loads of the corresponding realized OC, by using a uniform distribution. Similarly, a short-term predicted OC is generated by uniformly randomly changing the bus loads within 97% to 103% of the loads of the corresponding realized OC. After solving the power flows for each OC using the power flow and short circuit analysis tool (PSAT) [80], 481 OCs are generated for each of the three OC sets. Note that different from the day-ahead predicted OCs, the short-term predicted OCs and the realized OCs are time-stamped.

4.5.1.2 Critical Contingency Selection

A contingency list, which was created by the regional grid operator to account for possible outages of transmission lines, three-winding transformers and generators that could have significant impact, is used here. Specifically, the contingency list consists of 1 $N-4$ contingency, 8 $N-3$ contingencies, 172 $N-2$ contingencies, and 0 $N-1$ contingencies (i.e., no $N-1$ contingencies lead to insecure conditions). The power angle-based stability margin defined in (4.7) is used as the transient stability index. After performing transient security assessment by using TSAT for all realized OCs and adhering to the above security criteria, three $N-2$ contingencies which lead to transiently insecure cases are selected as the critical contingencies in the knowledge base. Each of the three $N-2$ critical contingencies is initiated by a "three-phase short circuit to ground" fault at a bus which is cleared after 5 cycles, by tripping a transmission line that connects the bus and by disconnecting a generator that will go out of step as a result of the line tripping.

4.5.1.3 Case Creation

Combining the three sets of generated OCs with their transient security classification decisions for the three critical contingencies, N=1443 cases are created for the initial knowledge base, for updating and for testing, respectively. Based on the interconnection structure of the 33 PMU buses, 799 numerical attributes are identified using the rules described in Section III-A; thus P=800. For each case, the values of the 799 numerical attributes are obtained from the power flow solutions. Then, the initial knowledge base is organized into an $N\times(P+1)$ array.

4.5.2 Offline Training

The initial knowledge base as an $N\times(P+1)$ array is first used by the CART algorithm to build the small DTs. Following the procedure described in Section III-D, it is found that J=2 and L=35 give the best results of V-fold cross validation. The first three small DTs built from the initial knowledge base are illustrated in Fig. 4.2. For comparison, a fully-grown single DT with pruning is also built, as illustrated in Fig. 4.1 (in order to give a concrete impression of DTs and small DTs used for DSA, Fig. 4.1 and Fig. 4.2 were presented in Section II).

4.5.3 Online DSA Simulation

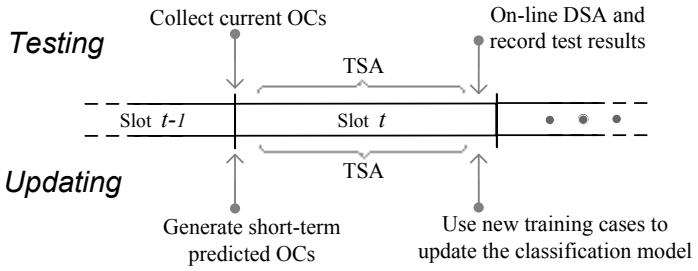


Figure 4.9: Flowchart for testing online DSA with periodic updating.

The online DSA is simulated iteratively on a slot-by-slot basis, as illustrated in Fig. 4.9. Generally, each slot spans M minutes. Since it is sufficient to perform security assessment of a short-term predicted OC for the three $N-2$ critical contingencies, M=1 is chosen here. In case of more critical contingencies or a larger test system, a longer slot can be chosen. In online DSA, a third scheme in which the classification model is obtained by boosting small DTs but updated by rebuilding is compared with the two aforementioned schemes.

4.5.3.1 OC variations in sub-period 12:00 Hrs-16:00 Hrs

In each slot of this sub-period, the $3M$ test cases created from the M realized OCs with time-stamps falling into this slot are collected, and then used as the present OCs for online DSA to assess the performance of the classification model updated so far. Meanwhile, another $3M$ new training cases created from the short-term predicted OC for the next slot are incorporated into the knowledge base to update the classification model.

4.5.3.2 Topology change in sub-period 16:00 Hrs-20:00 Hrs

At the peak hour 16:00 Hrs, a topology change is imposed on the test system, and assumed to last for the remaining hours of the day. Specifically, among the 178 contingencies that do not incur transient instability for all realized OCs, the contingency which has the least positive margin averaged over all realized OCs is chosen; as a result, a transmission line is removed and a generator is disconnected from the test system. Then, the new training cases and test cases during the latter sub-period are created using an approach similar to those used in the former sub-period, but by using a different system topology.

4.5.4 Test Results and Discussion

Throughout the entire horizon of the above online DSA simulations, the misclassification error rate and the computation time for updating in each slot are recorded and summarized in Table 4.2 and Fig. 4.10, respectively.

4.5.4.1 Classification Accuracy

As illustrated in Table 4.2, the two boosting-based schemes turn out to be more accurate than the single-DT-based scheme for both simulation sub-periods, and the performance of the proposed scheme is quite close to the scheme based on boosting small DTs with rebuilding.

4.5.4.2 Computation Requirement

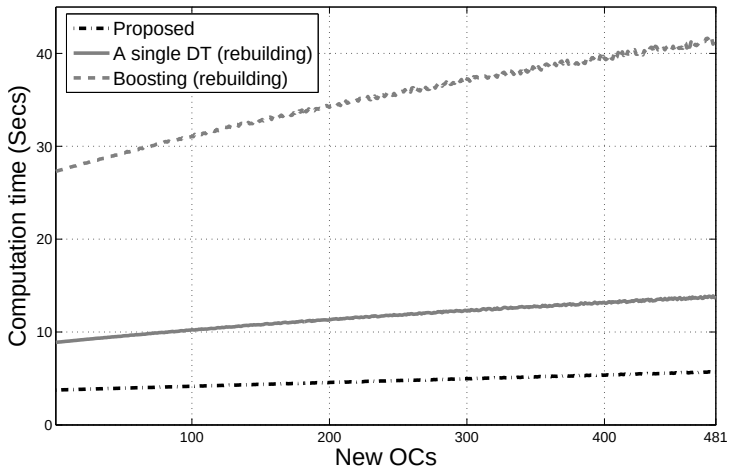


Figure 4.10: Computation time for updating/rebuilding (executed in MATLAB on a workstation with an Intel Pentium IV 3.20 GHz CPU and 4GB RAM).

The computation time required by updating the classification models using new OCs is illustrated in Fig. 4.10. It is clear that the proposed scheme requires the lowest computation time. Further, as the number of new OCs increases, the proposed scheme becomes less time-consuming than the other two schemes. The reason is that for each new OC, the two benchmark schemes rebuild DTs from scratch, while the graceful update of small DTs is carried out in the proposed scheme. Further, according to the CART algorithm [71], it is known that the sorting operation of the CART algorithm dominates the computational burden of DT building/rebuilding.

Table 4.2: Misclassification Error Rate of Online DSA

Scheme	Sub-period 12-16 Hrs			Sub-period 16-20 Hrs		
	secure cases	insecure cases	overall	secure cases	insecure cases	overall
Proposed	2.41%	1.03%	1.67%	2.54%	1.08%	1.74%
A single DT (rebuilding)	2.71%	1.80%	2.22%	2.26%	2.73%	2.5%
Boosting (rebuilding)	1.81%	1.03%	1.39%	2.26%	0.82%	1.5%

When updating small DTs, the sorting operation is skipped [75]. Therefore, the proposed scheme has a much lower computational burden.

4.6 Conclusion

In this study, a data-mining-based online DSA scheme is proposed to handle the OC variations and topology changes that are likely to occur during the operating horizon. The proposed scheme is applied to a practical power system, and the results of a case study demonstrate the performance improvement brought by boosting unpruned small DTs over a single DT. Compared to single DTs, the classification model obtained from ensemble DT learning often have higher accuracy, and lend themselves to cost-effective incorporation of new training cases. The results presented here also provide an insight into the possibilities of other ensemble DT learning techniques, e.g., random forest, in handling the challenges of online DSA.

Chapter 5

ROBUST ON-LINE DSA WITH MISSING SYNCHROPHASOR DATA

5.1 Introduction

DSA provides system operators with important information, e.g., transient security of a specific operating condition under various contingencies. Given a knowledge base, DTs can identify the attributes and the thresholds that are critical to assessing the transient performance of power systems [52, 54]. With the advent of synchrophasor technologies, a significant amount of effort has been directed towards *online* DSA, by using PMU measurements directly for decision making [55–57, 66]. Upon a disturbance, by applying pre-determined decision rules to the PMU measurements of critical attributes, DTs can give security classification decisions in real-time. In the online DSA schemes proposed in [55–57,66], security classification decision regions are first characterized by DTs in offline training, and then the real-time PMU measurements of critical attributes are used to obtain timely security classification decisions of the current OCs.

Previous studies on PMU measurement-based online DSA implicitly assume that WAMS provide reliable measurements. However, in online DSA, PMU measurements can become unavailable due to the unexpected failure of the PMUs or phasor data concentrators (PDCs), or due to loss of the communication links. Recently, it has been widely recognized that PMU failure can be an important factor that impacts the performance of WAMS. For example, AESO's newest rules on implementing PMUs [82] require that the loss or malfunction of PMUs, together with the cause and the expected repair time, has to be reported to the system operator in a timely manner. In the report [83], the deployment of redundancy is suggested by PMU manufacturers to reduce the impact of single PMU failure. Loss of PMUs has also

been taken into account when designing WAMS and PMU placement [84]. Moreover, the delivery of PMU measurements from multiple remote locations of power grids to monitoring centers could experience high latency when communication networks are heavily congested, which could also result in the unavailability of PMU measurements. Therefore, it is urgent to design DT-based online DSA approaches that are robust to missing PMU measurements.

Intuitively, one possible approach to handle missing PMU measurements is to estimate the missing values by using other PMU measurements and the system model. However, with existing nonlinear state estimators in SCADA systems, this approach may compromise the performance of DTs. First, the scan rate of SCADA systems is far from commensurate with the data rate of PMU measurements, and thus using estimated values from SCADA data may result in a large delay for decision making. Second, SCADA systems collect data from remote terminal units (RTUs) utilizing a polling approach. Following a disturbance, it is possible that some post-contingency values are used due to the lack of synchronization, which can lead to inaccurate security classification decisions of DTs. It is worth noting that future fully PMU-based linear state estimators [85] can overcome the aforementioned limitations; but this is possible only when there is a sufficient number of PMUs placed in system. With this motivation, data-mining based approaches are investigated in this chapter, aiming to use alternative viable measurements for decision making in case of missing data.

In DTs built by the CART algorithm [71], missing data can be handled by using *surrogate*. However, a critical observation in this study is that when PMU measurements are used as attributes, most viable surrogate attributes have low associations with the primary attributes. Clearly, the accuracy of DSA would degrade if surrogate is used. This is because a DT is essentially a sequential processing method,

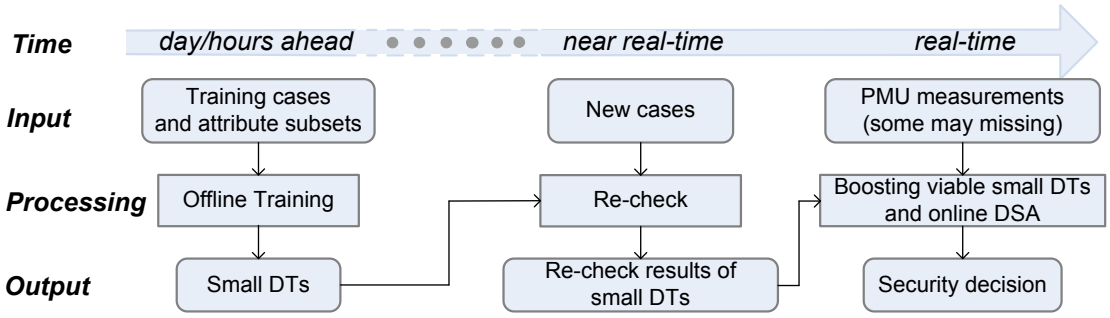


Figure 5.1: Three-stage ensemble DT-based approach to online DSA with missing PMU measurements

and thus the wrong decisions made in earlier stages may have significant impact on the correctness of the final decisions. Thus motivated, this chapter studies applying ensemble DT learning techniques, including random subspace methods and boosting, to improve the robustness to missing PMU measurements.

Aiming to develop a *robust* and *accurate* online DSA scheme, the proposed approach consists of three processing stages, as illustrated in Fig. 5.1. Specifically, given a collection of training cases, multiple small DTs are trained offline by using randomly selected attribute subsets. In near real-time, new cases are used to re-check the performance of small DTs. The re-check results are then utilized by a boosting algorithm to quantify the voting weights of a few *viable* small DTs (i.e., the DTs without missing data from their attribute subsets). Finally, security classification decisions of online DSA are obtained via a weighted voting of viable small DTs. More specifically, a random subspace method for selecting attribute subsets is developed by exploiting the locational information of attributes and the availability of PMU measurements. Conventionally, the availability of a WAMS is defined as the probability that the system is operating normally at a specified time instant [86]. In this study, the *availability* of PMU measurements is defined similarly, i.e., as the *probability* that PMU measurements are successfully collected and delivered to a monitoring center.

The developed random subspace method guarantees that a significant portion of small DTs are viable for online DSA with high likelihood. Further, a boosting algorithm is employed to assign the viable small DTs with proper voting weights that are quantified by using the results from performance re-check, leading to the high robustness and accuracy of the proposed approach in case of missing PMU measurements. The proposed approach is applied to the IEEE 39-bus system with 9 PMUs. Compared to off-the-shelf DT-based techniques (including random forests (RFs) with and without using surrogate), the proposed ensemble DT-based approach can achieve better performance in case of missing PMU measurements.

The rest of this chapter is organized as follows. An introduction to DTs with application to DSA is given in Section 5.2. Section 5.3 focuses on the random subspace method for selecting attribute subsets. The proposed three-stage approach is presented in detail in Section 5.4. A case study is discussed in Section 5.5. Finally, conclusions are given in Section 5.6.

5.2 Background on DTs

A *decision tree* is a tree-structured model that maps the measurements of the *attributes* $\mathbf{x}\in\mathcal{X}$ to a predicted value $\hat{y}\in\mathcal{Y}$ [71]. In a DT, a test on an attribute (thus called the primary attribute of the internal node) is installed at each internal node and decides which child node to drop the measurements into. Further, each leaf node of the DT is assigned a predicted value, and the measurements are thus labeled with the predicted value of the leaf node which it sinks into. The path from the root node to a leaf node specifies a decision region in the attribute space corresponding to that leaf node. Specifically, the length of the longest downward path from the root node to a leaf node is defined as the height of a DT.

5.2.1 Application of DTs in DSA

DTs with binary predicted values (i.e., $\mathcal{Y}=\{\pm 1\}$) are used in DSA. Specifically, $\hat{y}=+1$ represents that an OC is classified as *insecure* under a given contingency. The numerical attributes used by DTs in DSA include voltage magnitudes, voltage phase angles and power/current flows. Moreover, the index of contingencies is used as a categorical attribute. In DSA, a collection of training cases are first created by applying DSA packages (e.g., DSATools$^{\text{TM}}$ [80]) to N_{OC} known OCs for a given list of N_C contingencies. Then, the training cases $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$, where $N=N_{OC}\times N_C$, are used by the CART algorithm [71] to build a DT that fits the training data. Intuitively, if the DT fits the training data well and the new OCs in online DSA are similar to the OCs corresponding to the training cases, the trained DT can give accurate security classification decisions for the new OCs in online DSA.

In this study, *small* DTs, which have a small height J are used. Generally, a small DT could have lower accuracy than a fully-grown DT, but is less prone to overfitting when the training data is noisy [72], and multiple DTs are usually combined together to improve the classification accuracy.

5.2.2 Handling Missing Data by using Surrogate in DTs

A surrogate split at an internal node is the one that "mimics" the primary split most closely, i.e., gives the most similar splitting results for the training cases. Usually, the similarity is quantified by the *association* between the surrogate split and the primary split [71]. The significance of a surrogate split that has a high association (i.e., over 0.9) with the primary split is that the DT could still use the surrogate split at this internal node to give almost the same decisions when the PMU measurement of the primary attribute is missing.

The performance of surrogate in DT-based DSA is evaluated via a case study, in which a single DT is built by using the same knowledge base for voltage magnitude violation analysis as in [66]. It is observed that *co-located* attributes (i.e., the attributes measured by the same PMU) would often be unavailable at the same time when the PMU fails, which implies that co-located attributes cannot be used as surrogate for each other in online DSA. Therefore, a modified CART algorithm in which co-located attributes are excluded from surrogate searching is used to build a single DT and identify the surrogate attributes. The results regarding the performance of the surrogates identified by both the modified CART algorithm and the CART algorithm are given in Table 5.1. Two key observations are drawn. First, the results obtained by the modified CART algorithm suggest that all non-co-located surrogates have relatively low associations with the primary ones. The low association could be explained by the complex coupling structure of the attributes in power systems. According to the definition of surrogate, high association relies on the dependency between the surrogate and the primary attributes, i.e., the surrogate attribute gives similar decisions to the primary attribute on all the training cases regardless of any other attribute. However, in power systems, one attribute (i.e., voltage magnitude, voltage phase angle or power/current flow) is coupled with many other non-co-located attributes, as dictated by the AC power flow equations and the network interconnection structure. Second, it is observed in Table 5.1 that the surrogate attributes found by the CART algorithm are mostly co-located with the primary attributes. This observation signifies the redundancy between co-located attributes when used for splitting the training cases, and thus sheds lights on exploiting the locational information to create the attribute subsets, as described in Section III.

[1]Bus numbers are given in the subscripts of attributes, but are different from the real ones of the practical system. For example, $Q_{\{204,207\}}$ and $Q_{\{207,209\}}$ are co-located; they are measured by the same PMU at bus 207.

Table 5.1: A case study on the surrogates of DTs

node	primary attribute[1]	by modified CART		by CART	
		surrogate	assoc.	surrogate	assoc.
1	$V_{\{217\}}$	$V_{\{207\}}$	0.76	$V_{\{207\}}$	0.76
2	$Q_{\{204,207\}}$	$Q_{\{212,216\}}$	0.33	$Q_{\{207,209\}}$	0.50
3	$Q_{\{204,207\}}$	$V_{\{209\}}$	0.28	$Q_{\{207,209\}}$	0.64
4	$I_{\{211,204\}}$	$P_{\{008,011\}}$	0.62	$P_{\{209,211\}}$	0.83
5	$P_{\{210,201\}}$	$P_{\{211,062\}}$	0.87	$P_{\{231,201\}}$	0.87
6	$Q_{\{005,033\}}$	$Q_{\{801,999\}}$	0.71	$Q_{\{801,999\}}$	0.71
7	$P_{\{213,222\}}$	$Q_{\{207,211\}}$	0.85	$P_{\{222,223\}}$	0.85
8	$Q_{\{041,060\}}$	$I_{\{011,051\}}$	0.50	$I_{\{011,051\}}$	0.50
9	$P_{\{211,062\}}$	$P_{\{213,216\}}$	0.50	$I_{\{062,211\}}$	0.75
10	$P_{\{236,219\}}$	$Q_{\{230,052\}}$	0.42	$P_{\{236,207\}}$	0.68

5.2.3 Ensemble DT Learning

Ensemble DT learning techniques (bagging, random subspace methods, boosting, RF [74]) combine multiple DTs to obtain better prediction performance. Studies (e.g., [87]) have shown that using random subspace methods can lead to improved accuracy and generalization capability, if the DTs are trained from a variety of compact and non-redundant attribute subsets. Usually, the attribute subsets used by DTs are selected in a randomized manner. For example, in the random decision forest algorithm [88], each DT is built by using an attribute subset that is randomly selected from all possible candidate attribute subsets with equal weights. For online DSA, it is observed that additional system information on the attributes could be utilized to create and select the attribute subsets. First, the candidate attribute subsets could be significantly refined by exploiting the locational information of attributes. Further, by putting more weights on the attribute subsets that have higher availability when randomly selecting attribute subsets, the resulting small DTs would be more likely to be robust to possibly missing PMU measurements.

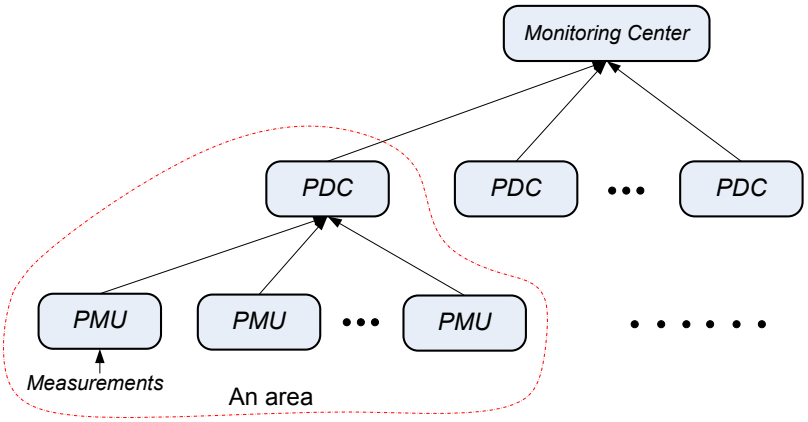


Figure 5.2: Wide area monitoring system consisting of multiple areas

5.3 Random Subspace Method for Selecting Attribute Subsets

A key step of the random subspace method is to identify a collection of candidate attribute subsets $\mathcal{S}$ and determine the weight $p_{\mathbf{s}}$ that dictates how likely a candidate attribute subset $\mathbf{s}\in\mathcal{S}$ is to be selected. In this study, by exploiting the locational information of attributes and the availability of PMU measurements, the random subspace method adheres to the following two guidelines:

- **G1**: Co-located attributes do not co-exist within an attribute subset.
- **G2**: The average availability of the selected attribute subsets should be sufficiently high.

Further, for a power system consisting of K areas, the corresponding WAMS is assumed to have a hierarchical architecture [89]. As illustrated in Fig. 5.2, each area of the power system has a PDC that concentrates the PMU measurements of this area and submits them to the monitoring center.

5.3.1 Candidate Attribute Subsets

The candidate attribute subsets are created based on the three following specific rules: **1**) Within a candidate attribute subset, all the attributes are from the same area. **2**) In area k (k=1,$\cdots$,K), three categories of pre-fault quantities measured by PMUs are used as the numerical attributes:

- Category 1: voltage magnitude V_i, for $i \in \mathcal{I}_k^{PMU}$;
- Category 2: active power flow P_{ij}, reactive power flow Q_{ij} and current magnitude I_{ij}, for $i \in \mathcal{I}_k^{PMU}$ and $j \in \mathcal{N}(i)$;
- Category 3: phase angle difference θ_{ij}, for $i,j \in \mathcal{I}_k^{PMU}$.

where $\mathcal{I}_k^{PMU}$ denotes the collection of the buses with PMU installation within area k, and $\mathcal{N}(i)$ denotes the collection of the neighbor buses of bus i. An attribute subset of area k is created by including one voltage or flow measurement from each bus $i \in \mathcal{I}_k^{PMU}$ and all phase angle difference measurements from this area. **3**) The index of contingencies is included as a categorical attribute in any attribute subset.

The criteria used in creating the attribute sets are elaborated below. By restricting the attributes of a subset to be the PMU measurements within the same area, the impact of some scenarios, i.e., when a PDC that concentrates PMU measurements within an area fails, is significantly reduced, since the small DTs using the PMU measurements from the other areas could still be viable. For a given bus, since Category 1 and Category 2 PMU measurements are co-located, it suffices to include only one of them in an attribute subset so that the redundancy within an attribute subset is minimal. Further, all measurable phase angle differences are included. This is because theoretical and empirical results (e.g., in [56]) suggest that

angle differences contain important information regarding the level of stress in OCs, and thus are more likely to be the attributes critical to assessing transient instability. It is also worth noting that the Category 2 attributes from two different buses are unlikely to be redundant, in the sense that they are the measurements from different transmission lines, given the fact that PMUs could provide power flow measurements and it is usually unnecessary to place PMUs at both ends of a transmission line to achieve the full observability of power grids.

For convenience, let $\mathcal{S}_k$ denote the collection of candidate attribute subsets of area k. Then, the size of $\mathcal{S}_k$ is given by

$$M_k = \prod_{i \in \mathcal{I}_k^{PMU}} (3\text{deg}(i) + 1), \tag{5.1}$$

where $\text{deg}(i)$ denotes the degree of bus i, i.e., the number of buses that connect with bus i. Then, $\mathcal{S}=\bigcup_{k=1}^{K}\mathcal{S}_k$ is the collection of candidate attribute subsets.

5.3.2 Randomized Algorithm for Selecting Attribute Subsets

It is plausible to develop the randomized algorithm so as to achieve maximum randomness of the selected attribute subsets by maximizing the entropy of the weight distribution $\{p_{\mathbf{s}},\mathbf{s}\in\mathcal{S}\}$. Without any other information of attribute, equal weights is usually used by existing random subspace methods (e.g., [90], [91]). Here, by adhering to guideline **G2**, an additional constraint is that the average availability of the randomly selected attribute subsets is above an acceptable level A_0. As a result, the weight distribution can be determined by solving the following problem:

$$\mathcal{P}_{\mathbf{s}} : \max_{\{p_{\mathbf{s}},\mathbf{s}\in\mathcal{S}\}} \sum_{\mathbf{s}\in\mathcal{S}} p_{\mathbf{s}} \text{log}_2 p_{\mathbf{s}}^{-1} \tag{5.2}$$

$$\text{s.t.} \quad \sum_{\mathbf{s}\in\mathcal{S}} p_{\mathbf{s}} A_{\mathbf{s}} \geq A_0, \tag{5.3}$$

$$\sum_{\mathbf{s}\in\mathcal{S}} p_{\mathbf{s}} = 1, \tag{5.4}$$

where $A_{\mathbf{s}}$ denotes the availability of an attribute subset $\mathbf{s}$. According to the rules for creating the candidate attribute subsets, it is easy to see that each of the attribute subsets of an area consists of exactly two measurements from each PMU within this area. Therefore, the availability of an attribute subset $\mathbf{s}$ of area k, which was formally defined in Section I as the probability that the measurements of $\mathbf{s}$ are successfully delivered to the monitoring center, equals that of the WAMS within area k, i.e.,

$$A_{\mathbf{s}} = A_k, \quad \forall \mathbf{s} \in \mathcal{S}_k. \tag{5.5}$$

In availability analysis of WAMS (e.g., in [86]), it is usually assumed that the availability of PMUs, PDCs and communication links are known (e.g., estimated from past operating data) and independent from each other. Under these assumptions, the availability of the WAMS within area k is given by:

$$A_k = \prod_{i\in\mathcal{I}_k^{PMU}} (A_i^{PMU} A_i^{link}) \tilde{A}_k^{PDC} \tilde{A}_k^{link}, \tag{5.6}$$

where A_i^{PMU}, A_i^{link}, $\tilde{A}_k^{PDC}$ and $\tilde{A}_k^{link}$ denote the availability of the PMU at bus i, the communication link from the PMU at bus i to the PDC, the PDC and the communication link from the PDC to the monitoring center, respectively. It is worth noting that (3) and (4) are derived for the case illustrated in Fig.2, and thus may not be directly applicable to the cases with measurement redundancy. For example, when multiple dual use PMU/line relays are utilized in substations, the availability of bus voltage phasor measurements can be enhanced. The procedure for analyzing the availability of WAMS in case of redundancy can be found in the literature (e.g., [92]).

By taking (5.5) into account, it follows that the solution to problem $\mathcal{P}_{\mathbf{s}}$ in (5.2) has the following property.

Proposition 5.3.1. *The optimal solution to $\mathcal{P}_{\mathbf{s}}$ in (5.2) takes the following form:*

$$p_{\mathbf{s}}^* = p_k^*/M_k, \quad \forall \mathbf{s} \in \mathcal{S}_k, \tag{5.7}$$

where M_k is the size of $\mathcal{S}_k$ as defined in (5.1), and $\{p_k^*, k=1,\cdots,K\}$ is the solution to the following problem:

$$\tilde{\mathcal{P}}_{\mathbf{s}} : \min_{p_1,\cdots p_K} \sum_{k=1}^{K} p_k log_2(p_k/M_k), \tag{5.8}$$

$$s.t. \quad \sum_{k=1}^{K} p_k A_k \geq A_0, \tag{5.9}$$

$$\sum_{k=1}^{K} p_k = 1. \tag{5.10}$$

Proof: Since $\mathcal{P}_{\mathbf{s}}$ maximizes a concave function with affine constraints, the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for a solution to be optimal. Therefore,

$$(1 + \ln p_{\mathbf{s}}^*)/\ln 2 - \lambda^* A_{\mathbf{s}} + \mu^* = 0, \quad \forall \mathbf{s} \in \mathcal{S} \tag{5.11}$$

where λ^* and μ^* are the KKT multipliers for the two constraints of $\mathcal{P}_{\mathbf{s}}$. Then, by taking the equality in (5.5) into account, it is easy to verify that $p_{\mathbf{s}}^*$ have the same value for all $\mathbf{s} \in \mathcal{S}_k$. Define $p_k = M_k p_{\mathbf{s}}$ for $\mathbf{s} \in \mathcal{S}_k$, then $\mathcal{P}_{\mathbf{s}}$ reduces to $\tilde{\mathcal{P}}_{\mathbf{s}}$.

The above result leads to the following implementation of the randomized algorithm, as summarized in Algorithm 2. Further, it is also observed from (5.11) that the attribute subsets which have higher availability are assigned higher weights.

Algorithm 2 Randomized algorithm for selecting an attribute subset

1: Calculate M_k and A_k according to (5.1) and (5.6), respectively, for $k = 1, \cdots, K$.
2: Find $\{p_k, k = 1, \cdots, K\}$ by solving $\tilde{\mathcal{P}}_{\mathbf{s}}$ in (5.8).
3: Select an area k among the K areas with weight p_k.
4: For the chosen area k, select an attribute subset s from $\mathcal{S}_k$ with weight $1/M_k$.

5.4 Proposed Approach for Online DSA with Missing PMU Measurements

First, L small DTs are trained offline by using randomly selected attribute subsets. In case of missing PMU measurements in online DSA, $\tilde{L}$ ($\tilde{L} \leq L$) viable small DTs are identified, and are assigned different voting weights. Specifically, the results of performance re-check in near real-time are utilized to quantify these voting weights. Finally, the security classification decisions for the new OCs in online DSA are obtained via weight voting of the $\tilde{L}$ viable small DTs.

5.4.1 Offline Training

Given a collection of training cases $\{\mathbf{x}_n, y_n\}_{n=1}^N$ and candidate attribute subsets $\mathcal{S}$, a primary objective of offline training is to obtain small DTs $\{h_1, \cdots, h_L\}$ so that the majority voting of them, i.e., $F_L(\mathbf{x}) = \sum_{l=1}^L h_l(\mathbf{x})$ could fit the training data. The iterative process to obtain a F_L is summarized in Algorithm 3. In the l-th iteration, a small DT h_l is first obtained by solving the following problem:

$$\mathcal{P}_{DT}^{(l)} : \min_{h_l} \frac{1}{N} \sum_{n=1}^{N} \mathbf{1}_{\{y_n \neq h_l(\mathbf{x}_n^l)\}}, \tag{5.12}$$

where $\mathbf{x}_n^l$ denotes the measurements of the attribute subset $\mathbf{s}_l$. It is well-known that the problem in (5.12) is NP-complete [93]. Here, the CART [71] algorithm is employed to find a sub-optimal DT, by using misclassification error rate as the splitting cost function. It is clear from (5.12) that equal weights, i.e., $\frac{1}{N}$, are assigned to all training data. When historical data that identifies potential weak spots of the

system is available, these data can be integrated by assigning higher weights, and by replacing $\frac{1}{N}$ with unequal data weights.

Algorithm 3 Offline training using the random subspace method

1: **Input:** Training cases $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$, $\varepsilon_0 \in (0, 1)$.
2: **Initialization:** $F_0 = \mathbf{0}$.
3: **for** $l = 1 \rightarrow L$ **do**
4: Select an attribute subset $\mathbf{s}_l$ by using Algorithm 2.
5: Find a small DT h_l by solving $\mathcal{P}_{DT}^{(l)}$ in (5.12) using the CART algorithm.
6: $F_l \leftarrow F_{l-1} + h_l$.
7: **end for**

5.4.2 Near Real-time Performance Re-check

In near real-time, a more accurate prediction of the imminent OC in online DSA can be made. Then, a collection of new cases $\{\tilde{\mathbf{x}}_n, \tilde{y}_n\}_{n=1}^{\tilde{N}}$ are created in a similar manner to that in offline training and used to re-evaluate the accuracy of the L small DTs. The re-check results are then utilized by the boosting process in online DSA. In case of variations between the OCs used in offline training and the new OCs in online DSA, near real-time re-check is also a critical step to make sure that the small DTs still work well.

5.4.3 Online DSA

The results of near real-time re-check $\{\tilde{h}_l(\tilde{\mathbf{x}}_n^l), \tilde{y}_n\}_{n=1}^{\tilde{N}}$, $\forall l{=}1,\cdots,\tilde{L}$ are utilized to choose a few viable small DTs to be used in online DSA and calculate the corresponding voting weights via a process of boosting small DTs. In order to make best use of existing DTs, the viable small DTs in online DSA include the small DTs without any missing PMU measurement and non-empty degenerate small DTs.

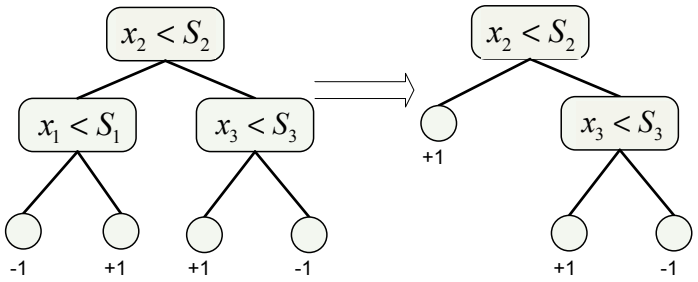


Figure 5.3: Degeneration of a small DT as a result of missing PMU measurements of attribute x_1 when node (x_1<S_1) is originally assigned +1.

5.4.3.1 Degenerate Small DTs

A *degenerate* small DT is obtained by collapsing the subtree of an internal node with missing PMU measurement into a leaf node. Specifically, a small DT degenerates to a non-empty tree if the PMU measurements used by the internal nodes other than the root node are missing, an example of which is illustrated in Fig. 5.3. Further, since each internal node of the original small DT is also assigned a decision in building the DT, the new leaf node of the degenerate small DT is assigned the same decision as the original internal node. Therefore, for a non-empty degenerate small DT, the re-check results on the $\tilde{N}$ new cases could be easily obtained.

5.4.3.2 Weighted Voting of Viable Small DTs

Let $\tilde{\mathcal{H}}$ be the collection of viable small DTs. Then, weighted voting of the viable small DTs in $\tilde{\mathcal{H}}$ is utilized to obtain the security classification decisions of online DSA, due to the following two reasons. First, in case that some small DTs degenerate to empty trees and the accuracy of non-empty degenerate small DTs degrades, weighted voting could improve the overall accuracy compared to majority voting, provided that the voting weights are carefully assigned based on the re-check results of the viable small DTs. Second, even though all the small DTs are viable, choosing the small DTs with

proper voting weights based on their accuracy can still be a critical step to guarantee accurate decisions. This is because small DTs trained offline fit the training cases that are created based on day-ahead prediction, while the re-check results on the $\tilde{N}$ new cases contain more relevant information on assessing the security of the imminent OCs in online DSA.

In the proposed approach, weighted voting of small DTs in $\tilde{\mathcal{H}}$ is implemented via a boosting process. Following the method in [66], initially with $\tilde{F}_0$ as a zero function, a small DT $\tilde{h}_l \in \tilde{\mathcal{H}}$ is first identified and added to $\tilde{F}_{l-1}$, i.e.,

$$\tilde{F}_l = \tilde{F}_{l-1} + a_l \tilde{h}_l, \tag{5.13}$$

iteratively for l=1,2,$\cdots$,$\tilde{L}$, so that the cost function, i.e.,

$$\hat{C}(\tilde{F}_{\tilde{L}}) = \frac{1}{\tilde{N}} \sum_{n=1}^{\tilde{N}} \log_2(1 + \mathrm{e}^{-\tilde{y}_n \tilde{F}_{\tilde{L}}(\tilde{\mathbf{x}}_\mathbf{n})}), \tag{5.14}$$

is minimized in a gradient descent manner. In the boosting process, $\tilde{h}_l$ is identified by solving the following problem:

$$\tilde{\mathcal{P}}_{DT}^{(l)} : \min_{h_l \in \tilde{\mathcal{H}}} \frac{1}{\tilde{N}} \sum_{n=1}^{\tilde{N}} w_n^{(l)} \mathbf{1}_{\{\tilde{y}_n \neq \tilde{h}_l(\tilde{\mathbf{x}}_n^l)\}}, \tag{5.15}$$

and the data weights and voting weight are given by

$$\begin{cases} w_n^{(l)} = \frac{1}{1+\mathrm{e}^{\tilde{y}_n \tilde{F}_{l-1}(\tilde{\mathbf{x}}_n)}} & n = 1, \cdots, \tilde{N} \\ a_l = \underset{a \in \mathcal{R}^+}{\operatorname{argmin}}\, g_l(a) \end{cases} \tag{5.16}$$

where $g_l(a) \triangleq \hat{C}(\tilde{F}_{l-1} + a\tilde{h}_l)$. Boosting viable small DTs in online DSA is summarized in Algorithm 4.

Algorithm 4 Boosting viable small DTs for online DSA

Input: Re-check results $\{\tilde{h}_l(\tilde{\mathbf{x}}_n^l), \tilde{y}_n\}_{n=1}^{\tilde{N}}$, $\forall l = 1, \cdots, \tilde{L}$,
Initialization: $\tilde{F}_0 = \mathbf{0}$.
for $l = 1 \rightarrow \tilde{L}$ **do**
    Calculate the data weights according to (5.16).
    Find a small DT $\tilde{h}_l$ from viable DTs by solving $\tilde{\mathcal{P}}_{DT}^{(l)}$ in (5.15).
    Calculate the voting weight a_l according to (5.16).
    $\tilde{F}_l \leftarrow \tilde{F}_{l-1} + a_l \tilde{h}_l$.
end for

5.4.4 Further Discussion

Through detailed complexity analysis, it is shown that the low computational complexity of the online processing renders that the time criticality of online DSA would not be compromised when the proposed approach is used. Specifically, the computationally intensive part of the online processing stage is the boosting process that consists of calculating the data weights $w_n^{(l)}$, solving $\tilde{\mathcal{P}}_{DT}^{(l)}$ and calculating the voting weights a_l of small DTs. According to (5.16), calculating the data weights requires evaluating $\tilde{F}_l$ for the new cases, which could be easily obtained from the re-check results of the small DTs. Therefore, it is easy to see that the complexity in calculating the data weights is $\mathcal{O}(\tilde{N})$. Solving $\tilde{\mathcal{P}}_{DT}^{(l)}$ boils down to searching for the small DT in $\tilde{\mathcal{H}}$ that has the least weighted misclassification error. Since the re-check results of the small DTs in $\tilde{\mathcal{H}}$ for the new cases are already known, the optimal small DT could be found by comparing the weighted misclassification errors of the small DTs in $\tilde{\mathcal{H}}$. Therefore, the complexity in solving $\tilde{\mathcal{P}}_{DT}^{(l)}$ is $\mathcal{O}(\tilde{L}\tilde{N})$. In the l-th iteration of the boosting process, the voting weight is obtained by minimizing $g_l(a)$. It is easy to verify that $g_l'(0)<0$ and $g_l''(a)>0$ holds for $a \in \mathcal{R}^+$. Therefore, $g_l(a)$ has a unique minimum in R^+ that could be found by using standard numerical methods (e.g., Newton's methods). Further, since $g_l(a)$ is convex, standard numerical methods could find the minimum in a few iterations. In each iteration, $\tilde{F}_{l-1}+a\tilde{h}_l$ needs to be evaluated for all the $\tilde{N}$

new cases. Therefore, the complexity in calculating the voting weight for a small DT is $\mathcal{O}(\tilde{N})$. Summarizing, the overall computational complexity of the boosting process is $\mathcal{O}(\tilde{L}^2\tilde{N})$.

The proposed approach above relates to that in [66] in the following sense: small DTs are utilized in both approaches; new cases are used in near real-time for accuracy guarantee by both approaches; the security classification decisions of online DSA are both obtained via a weighted voting of small DTs. However, the two approaches are tailored towards different application scenarios. The approach proposed here is more robust to missing PMU measurements, while the approach in [66] could give accurate decisions with less effort in offline training when the availability of PMU measurements is sufficiently high. The major differences of the two approaches are outlined as follows. First, the small DTs in the proposed approach are trained by using attribute subsets for robustness, whereas the entire set of attributes is used in [66]. Second, the usage of new cases in near real-time is different. In [66], the new cases are used to update the small DTs, whereas in the proposed approach, the new cases are only used to re-check the performance of viable small DTs so as to quantify the voting weights.

5.5 Case Study

5.5.1 Test System

The IEEE 39-bus system [78] is used as the test system which contains 39 buses, 10 generators, 34 transmission lines and 12 transformers. Particularly, G1 represents the aggregated generation from the rest of eastern interconnection [78]. In this case study, the test system is assumed to consist of three areas. The three areas together with the PMU placement are illustrated in Fig. 5.4. It is worth noting that the PMU placement guarantee the full observability of the test system when zero-injection buses

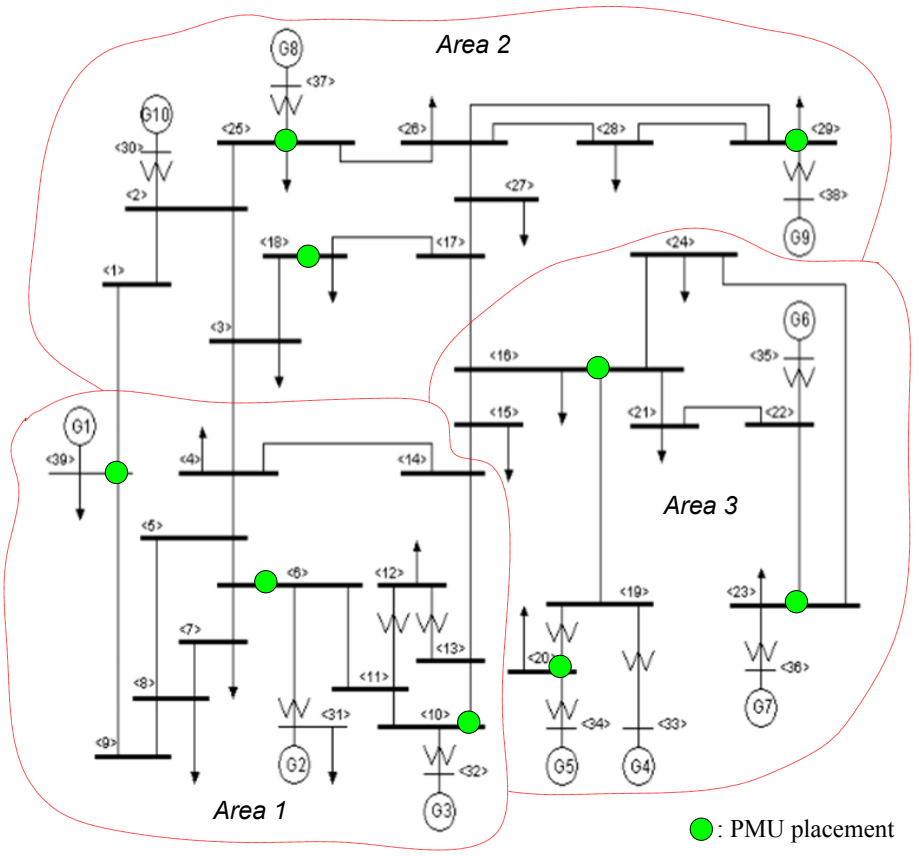


Figure 5.4: The IEEE 39-bus system in three areas and PMU placement

are taken into account.

5.5.2 Knowledge base

The knowledge base only consists of the OCs that are both pre-contingency secure and $N-1$ secure. The cases in the knowledge base are created from the combinations of the "PMU measurements" of the OCs and their transient security classification decisions for a few selected $N-2$ contingencies. In this case study, the power flow solutions of an OC are used as the "PMU measurements."

5.5.2.1 OC Generation

The OC given in [78] is used as the base OC. Following the method in [57], more OCs are generated for offline training, by randomly changing the bus loads (both active and reactive) within 90% to 110% of their original values in the base OC; for the OCs generated for near real-time re-check and online DSA test, the bus loads varies from 97% to 103% of their original values in the base OC. The rationale for the above percentage values is that offline training is usually carried out day/hours ahead, and thus the predicted OCs can have a larger prediction error than those in near real-time. The power flows of each generated OC are solved using the PSAT software [80], followed by a limit check such that the generated OCs with any pre-contingency overloading or voltage/angle limit violations are excluded from the knowledge base.

5.5.2.2 Contingencies

The loss of any of the 46 components (i.e., the 34 transmission lines, the 9 generator-transformer pairs and the 3 transformers at (11,12), (12,13) and (19,20)) is considered as an $N-1$ contingency. Due to the large number of possible $N-2$ contingencies, only a few of them are selected. Intuitively, a severe impact on the security of power systems is more likely if a second component gets overloaded after the loss of the first component. As such, the $N-2$ contingencies are selected in the following manner. First, each of the aforementioned 46 components is removed from the test system. Then, power flows are re-solved and limit check is re-run for the base OC using PSAT. The first removed component together with any overloaded component are regarded as the removed pair of an $N-2$ contingency. As a result, 15 pairs are identified, as listed in Table 5.3.

Table 5.2: Data used by Algorithm 1 for the system in Fig. 5.4.

Area	Placement	Number of attributes			M_k	A_k	p_k
		Cat. 1	Cat. 2	Cat. 3			
1	8, 13, 39	3	24	3	700	b	0.28
2	18, 25, 29	3	24	3	700	b	0.28
3	16, 20, 23	3	30	3	1120	b	0.44

5.5.2.3 Transient Security Assessment

TSAT [80] is used to assess the transient performance of the OCs that are pre-contingency secure. To create a contingency in TSAT, the "three-phase short circuit to ground" fault is applied at either of the two terminal buses of the first removed component with a primary clearing time of 4 cycles. Therefore, 92 $N-1$ contingencies and 30 $N-2$ contingencies are created. The "power angle-based stability margin" defined in TSAT [80] is used as the stability index.

5.5.3 Performance Evaluation

Three other approaches are used as benchmarks, including a single DT using surrogate, an RF using surrogate, and an RF without using surrogate. Following [91], unpruned DTs are used in RFs; in RFs, all training cases are used to build a single DT; in each split of DTs, a number of $\log_2 P + 1$ attributes are randomly selected (where P=96 according to Column 3 of Table III); the optimal number of DTs in the forest is determined through out-of-bag validation [91]. Specifically, for the former two benchmark approaches, surrogate attributes are obtained from those which are not co-located with the primary attributes; for the third benchmark approach, degenerated DTs are used.

5.5.3.1 Attribute Subsets

The hypothetical WAMS for the test system has a hierarchical architecture similar to that in Fig. 5.2. Based on the evaluation results of the reference [94], it is assumed that all the PMUs have the same availability a ($a\in$ [0.979975,0.998920]), and all the communication links from PMUs to PDC have the same availability A^{link}=0.999. Further, the availability of the PDC and the communication link from the PDC to the monitoring center is assumed to be 1. Let b=$(0.999a)^3$, and thus $b\in$ [0.938299,0.993776]. Then, it follows that when $A_0 \leq b$, the solution to $\tilde{\mathcal{P}}_{\mathbf{s}}$ in (6) exists, as given in Table 5.2. In what follows, the data in Table 5.2 is explained in detail. Specifically, column 2 provides the indices of PMU buses, which can also be seen from Fig.4. Column 3 contains the number of attributes for the three categories defined in Section III.A. Take area 1 for example, there are 3 voltage magnitude attributes, 24 transmission line (including power flow and current magnitude) attributes, and another 3 attributes from voltage phase angle difference. Given the system topology and availability information, M_k in column 4 and A_k in column 5 are calculated using (1) and (4), respectively. Then, p_k is obtained by solving (6).

5.5.3.2 Offline Training

N_{OC}=200 generated OCs which are both pre-contingency and $N-1$ contingency secure are used for offline training. Combining the generated OCs with their transient security classification decisions for the N_C=30 selected $N-2$ contingencies, are used to generate the N=6000 cases in the knowledge base. The size and the number of small DTs are determined by bias-variance analysis [68] and v-fold cross validation [56]. In this case study, L=40 and J=3 are used by the proposed approach; 45 DTs are used in the two RF-based approaches.

Table 5.3: The first and second removed components of the selected $N-2$ contingencies

line(4,14), line(6,11)	line(6,11), line(4,14)	line(6,11), line(13,14)
line(6,11), line(10,13)	line(10,11), line(10,13)	line(10,13), line(6,11)
line(10,13), line(10,11)	line(13,14), line(6,11)	line(13,14), line(10,11)
line(16,21), line(23,24)	line(21,22), line(23,24)	line(21,22), line(22,23)
line(21,22), line(16,24)	line(23,24), line(16,21)	line(23,24), line(21,22)

5.5.3.3 Near Real-time Re-check

By following the procedure described in Section V.B, 100 OCs are generated for performance re-check. The DTs trained offline are applied to the new cases; the classification results are compared with the actual security classification decisions of the new cases. Then, these re-check results are used by Algorithm 4 to quantify the voting weights of DTs.

5.5.3.4 Online DSA Test

Another 100 OCs are generated for testing, by following the procedure described in Section V.B. Recall that the availability of the PDCs and the communication links connecting PDCs is 1, and then it can be seen from Fig. 5.2 and Fig. 5.4 that the total number of failure scenarios can be reduced to 512 (2^9, since there are 9 pairs of PMUs and links). Online DSA test is repeated for all failure scenarios, by identifying the missing PMU measurements and viable small DTs, calculating the voting weights of viable small DTs, and evaluating the misclassification error rate. The overall misclassification error of online DSA is calculated by:

$$\bar{\mathrm{e}}(\tilde{F}) = \sum_{k=1}^{512} \mathrm{Prob}(\Omega(k))\mathrm{e}(\tilde{F}|\Omega(k)), \tag{5.17}$$

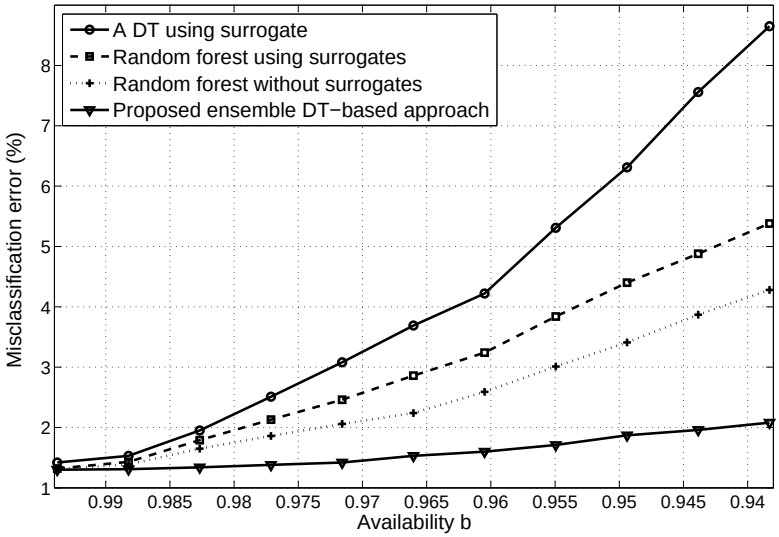


Figure 5.5: Performance of online DSA in case of missing PMU measurements

Table 5.4: Misclassification error rate when b=0.94.

Scheme	Secure cases	Insecure cases	overall
A single DT	8.92%	8.63%	8.74%
RF with surrogates	5.14%	5.81%	5.33%
RF without surrogates	4.52%	4.17%	4.28%
Proposed	2.36%	1.92%	2.03%

where, $\Omega(k)$ denotes the k-th failure scenario, Prob($\Omega(k)$) denotes the probability for $\Omega(k)$ to happen, which can be easily calculated by using the assumed availability, and e($\tilde{F}|\Omega(k)$) denotes the misclassification error rate of $\tilde{F}$ in the failure scenario $\Omega(k)$ (e($\tilde{F}|\Omega$) is set to be 1 when all PMUs fail). The test is performed for various values of b, and the test results are illustrated in Fig. 5.5. It is observed that the benchmark approaches are comparable to the proposed approach only around b=1. However, the gaps become more significant as b decreases. More specifically, the misclassification error rates when the availability b=0.94 are shown in Table 5.4, for "Secure" and "Insecure" cases respectively.

5.5.3.5 Impact of Measurement Noise

In reality, PMU data can contain measurement noise. Following the approach in [95], numerical experiment is carried out to study the impact of measurement noise on the performance of the proposed approach.

For convenience, let $V\angle\theta_V$ and $I\angle\theta_I$ denote a voltage phasor and a current phasor, respectively; let $\tilde{V}\angle\tilde{\theta}_V$ and $\tilde{I}\angle\tilde{\theta}_I$ be the corresponding measurement. For PMUs complying with IEEE C37.118 standard [96], the PMU measurements should have a total vector error (TVE) less than 1%, i.e.,

$$|\frac{\tilde{V}\angle\tilde{\theta}_V - V\angle\theta_V}{V\angle\theta_V}| < 1\% \tag{5.18}$$

$$|\frac{\tilde{I}\angle\tilde{\theta}_I - I\angle\theta_I}{I\angle\theta_I}| < 1\% \tag{5.19}$$

Let n_V and n_I denote the measurement noise, respectively. In order to obtain PMU measurements that comply with the above specifications, n_V and n_I are randomly generated, by using the following density functions (note that other density functions can be also used) that are properly scaled and truncated from the standard complexity Gaussian distribution:

$$f(n_V) = \begin{cases} \frac{9}{\pi(1-e^{-9})10^{-4}V^2} e^{-\frac{9|n_V|^2}{10^{-4}V^2}} & if \ \ |n_V| \leq 10^{-2}V \\ 0 & o.w. \end{cases} \tag{5.20}$$

$$f(n_I) = \begin{cases} \frac{9}{\pi(1-e^{-9})10^{-4}I^2} e^{-\frac{9|n_I|^2}{10^{-4}I^2}} & if \ \ |n_I| \leq 10^{-2}I \\ 0 & o.w. \end{cases} \tag{5.21}$$

Then, it is clear that all noisy measurements have TVE not more than 1%, and are complex Gaussian distributed within their support. The generated random mea-

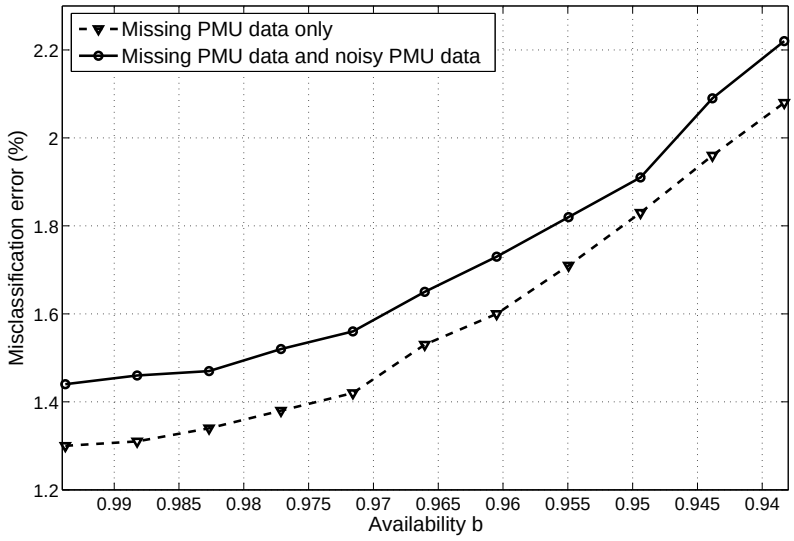


Figure 5.6: Impact of measurement noise.

surement noise is added to the both training and testing data. The test results are provided in Fig. 5.6.

5.6 Conclusion

A data-mining approach has been proposed to mitigate the impact of missing PMU measurements in online DSA. In particular, the various possibilities of missing PMU measurements in online DSA can make off-the-shelf DT-based techniques (e.g., a single DT, RF) fail to deliver the same performance as expected. The proposed ensemble DT-based approach exploits the locational information and the availability information of PMU measurements in randomly selecting attribute subsets, and utilizes the re-check results to re-weight the DTs in the ensemble. These special treatments developed from a better characterization of power system dynamics guarantee that the proposed approach can achieve better performance than directly applying off-the-shelf DT-based techniques.

Chapter 6

A DEPENDENCY GRAPH APPROACH FOR FAULT DETECTION AND LOCALIZATION USING SYNCHROPHASOR DATA

6.1 Introduction

One of the primary challenges on securing power systems has been the early detection and localization of fault events so as to mitigate their impacts on the overall stability of the power grid. It has been well recognized that a lack of "situational awareness" is often the main reason for large-scale fault events that begin with failures in one area and eventually propagate to other regions, e.g., the 2003 blackout that spread across the eastern United States and Canada. However, today's power systems are not equipped with adequate fault detection and localization mechanisms against various malicious attacks and natural physical events [97]. There is therefore an urgent need to enhance the situational awareness, so that corrective actions can be taken promptly to avoid cascading events.

Traditionally, in SCADA systems, critical buses in the transmission system are monitored using the data measured at RTUs. These measurements could provide only a snapshot of power systems, i.e., the static or quasi-dynamic status. In contrast, WAMS [98], in which the measurements are made at a much finer granularity, is designed to enhance the system operator's real-time situational awareness. In a WAMS, the current, voltage and frequency phasor at the monitored buses are collected by PMUs, together with the time stamps provided by GPS. After processing these highly synchronized measurements, the system operator is able to observe not only the steady-state, but also the dynamic state of the entire power system. The availability of synchronized measurements has enabled a new level of situational awareness. Indeed, recent years have witnessed a surge of various WAMS-based appli-

cations, e.g., voltage and frequency instability analysis [56], fault diagnosis [99–101], and the detection of cascading failures [102].

It is known that the fault localization of transmission lines is challenging [97], due to the massive scale, global coupling and time-varying system states. Specifically, the complexity of fault localization, if not taken care of carefully, can be extremely high, since it involves a large volume of PMU measurements from a massive-scale transmission system. Moreover, in power systems, given the bus injections, the voltage and phase angle of a bus are determined by the states of neighboring buses and the electrical connectivity, as described by Kirchoff's law. The inherent dependency between the states of buses dictates that the PMU measurements are spatially correlated and coupled. Accordingly, anomalies in the PMU measurements at the buses of a transmission line might be the consequence of failures at other lines. Further, due to stochastic events in power systems, the abnormal trajectories of PMU measurements might be the result of a sudden change in the power injections from load or generation, other than the failure of transmission lines. In light of the stochastic nature of power systems, bus injections and branch flows could be volatile across various time scales, which could be even more phenomenal in smart grids that are supposed to integrate a large number of distributed generations and renewable energy resources.

A main objective of this study is to devise a decentralized fault localization scheme that can capture the latent dependency and the uncertainty of power systems. We note that Markov random fields (MRFs) and graphical models [103] offer a convenient platform to characterize the dependency between a large number of random variables, and thus, have found various applications in many fields (see [104–106] and the references therein). In particular, based on the Hammersley-Clifford Theorem [107], graphical models have shown great potential for designing decentralized algorithms for large scale inference problems.

In this chapter, we will explore decentralized network inference using graphical models to address the following challenges in the fault detection and localization of transmission lines: 1) the computational complexity of large scale inference problems; 2) the global coupling and correlation between the PMU measurements; and 3) the uncertainty of system states caused by stochastic load and generation.

6.2 Dependency Graph Models for PMU Data

6.2.1 Background on Dependency Graph

A *random field* $\mathbf{X}$ on a finite set S with p sites is a collection of random variables $X_s, s \in S$, with values in *state space* Λ, and a *configuration* $\mathbf{x} \in \Lambda^S$ on $\mathbf{X}$ is a collection of the values of the random variables. For a site $s \in S$ in a *Markov random field*, its *neighborhood* N_s is a subset of S such that the following Markov property holds [108]:

$$X_s \perp \mathbf{X}_{S \setminus \{s \cup N_s\}} | \mathbf{X}_{N_s}, \tag{6.1}$$

i.e., given $\mathbf{X}_{N_s}$, X_s is conditionally independent of the other random variables. It is worth mentioning that the neighborhood of X_s is non-trivially not unique, and N_s is said to be *minimal* [108] if none of $X_t, t \in N_s$, satisfies the pairwise Markov property:

$$X_s \perp X_t | \mathbf{X}_{S \setminus \{s,t\}}. \tag{6.2}$$

In what follows, we use N_s to denote the minimal neighborhood of site s. Note that this neighborhood relation is symmetrical, i.e., $t \in N_s$ if and only if $s \in N_t$.

The *dependency graph* $G = (S, E)$ of MRF $\mathbf{X}$ is obtained by placing an edge $\{s, t\} \in E$ between each pair of sites s and t that are in each other's minimal neighborhood. We use the standard notation in graph theory, namely $s \sim t$, to denote this adjacency relation in undirected graph, otherwise it is $s \not\sim t$. A *clique* c in G

is a subset of S, either with cardinality $|c| = 1$ or such that any two sites in c are neighbors in the dependency graph. Let $\mathcal{C}$ denote the collection of cliques in G.

According to the Hammersley-Clifford theorem [107], under the positiveness condition, i.e., $P_{\mathbf{X}}(\mathbf{x}) > 0$ for any configuration $\mathbf{x} \in \Lambda^S$, the probability distribution of MRF $\mathbf{X}$ can be expressed in terms of potential functions. A *potential function on clique* c is a measurable mapping $V_c : \Lambda^S \to R$, that depends only on the variables $\{X_s, s \in c\}$, i.e., if two configurations $\mathbf{x}_1, \mathbf{x}_2 \in \Lambda^S$ agree on the values of the sites contained in c, then $V_c(\mathbf{x}_1) = V_c(\mathbf{x}_2)$. An *irreducible potential function* V_c is either zero everywhere on Λ^S or could not be represented as a sum of non-zero potential functions on other cliques that are the subsets of c [108].

A *Gaussian Markov random field* is a MRF that has probability distribution $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{J}^{-1})$, where $\mathbf{J}$ is the information matrix of the GMRF. Without loss of generality, we consider zero-mean GMRFs, and the diagonal entries of $\mathbf{J}$ are normalized to 1. It is clear that the positiveness condition is satisfied by GMRFs, accordingly, the probability distribution of GMRF $\mathbf{X}$ can be expressed in terms of non-zero irreducible potential functions on the singleton and pairwise cliques [109]:

$$P_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \exp\left\{ -\sum_{i \in S} V_{\{i\}}(x_i) - \sum_{(i,j) \in E} V_{\{i,j\}}(x_i, x_j) \right\}, \tag{6.3}$$

where, the partition function is given by

$$Z = (2\pi)^{p/2} |\mathbf{J}|^{-1/2}, \tag{6.4}$$

and potential function on singleton clique $\{i\}$ is given by

$$V_{\{i\}}(x_i) = \frac{1}{2} \mathbf{J}_{ii} x_i^2, \tag{6.5}$$

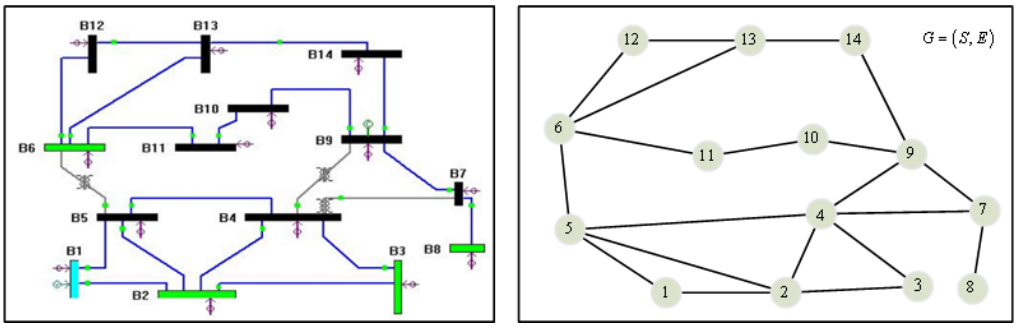


Figure 6.1: The IEEE 14-bus system and the dependency graph of the phase angles

and potential function on pairwise clique $\{i, j\}$ is given by

$$V_{\{i,j\}}(x_i, x_j) = \mathbf{J}_{ij} x_i x_j. \tag{6.6}$$

One key observation from (6.5) and (6.6) is that each site or edge in the dependency graph of a GMRF corresponds to a non-zero entry of the information matrix, vice versa, i.e., for GMRF $\mathbf{X}$, $i \sim j$ if and only if $\mathbf{J}_{ij} \neq 0$.

6.2.2 Gaussian Markov Random Field for Phase Angles

The DC power flow model [110] is often used for the analysis of power systems in normal steady-state operations, where the power flow on the transmission line connecting bus i to bus j is given by

$$Z_{ij} = b_{ij}(X_i - X_j), \tag{6.7}$$

where X_i and X_j denote the phase angles at bus i and bus j, respectively, and b_{ij}, the inverse of line inductive reactance, measures the electrical connectivity between bus i and bus j. Further, the power injection to bus i, which is equal to the algebraic sum of the power flowing away from bus i, is given by

$$Z_i = \sum\nolimits_{j \neq i} Z_{ij} = \sum\nolimits_{j \neq i} b_{ij} \, (X_i - X_j). \tag{6.8}$$

It follows that the phase angle at bus i can be represented as

$$X_i = \sum\nolimits_{j \neq i} r_{ij} X_j + \beta Z_i, \tag{6.9}$$

with $\beta \triangleq \left(\sum_{j \neq i} b_{ij} \right)^{-1}$ and $r_{ij} \triangleq \beta b_{ij}$.

Existing probabilistic power flow approaches [111] usually model the load flows at buses as random variables, to account for the random disturbances and uncertainty of various load profiles. Some of these approaches [111,112] model the aggregated load flow injection at buses as Gaussian random variables. Following these approaches, we assume that during the observation window of fault diagnosis applications, the flow injection originating from the aggregated load requests of a large number of users can be well approximated by Gaussian random variables.

According to [113], branch flows can be expressed as linear combinations of the power injection at buses. It follows that branch flow Z_{ij} could also be modeled as a Gaussian random variable. Since the phase difference of buses would not be too large in order to drive branch flows [114], the linear relationship shown in (6.7) implies that the difference of phase angles across a bus can be approximated by a Gaussian random variable truncated within $[0, 2\pi)$. In power systems, since the phasor is fixed at the slack bus, we assume that, under steady-state, the phase angles at non-slack buses can be approximately modeled as Gaussian random variables.

In a nutshell, based on (6.9), the conditional distribution of X_i can be specified in the form of a conditional auto-regression (CAR) model [115], i.e.,

$$X_i \,|\mathbf{X}_{-i} \sim \mathcal{N}\left(\sum\nolimits_{j \neq i} r_{ij} x_j, 1 \right), \tag{6.10}$$

with $\mathbf{X}_{-i}$ denoting the random variables on $S \backslash \{i\}$. Here, we abuse the notation and

use $X_i|\mathbf{X}_{-i}$ to denote the conditional distribution of X_i given $\{X_j = x_j,\ j \neq i\}$, and the distribution is normalized to highlight the conditional correlations. It is shown in [115] that under the condition that $\mathbf{I} - \mathbf{R}$ is positive definite, GMRF $\mathbf{X}$ follows the joint distribution $\mathcal{N}(\mathbf{0}, (\mathbf{I} - \mathbf{R})^{-1})$, with $\mathbf{R} \triangleq [r_{ij}]$ as the matrix consisting of conditional correlation coefficients. Note that for each X_i, its conditional correlation coefficients $\{r_{ij}, j \neq i\}$ are proportional to $\{b_{ij}, j \neq i\}$. Through building the dependency graph of phase angles on the topology of power systems, as illustrated in Fig. 6.1, we observe that, similar to the susceptance matrix $\mathbf{B} \triangleq [b_{ij}]$, the conditional correlation matrix $\mathbf{R}$ also reflects the electrical distance between buses. Intuitively, *the change of the electrical connectivity of buses would result in a different conditional correlation coefficient between phase angles; particularly, r_{ij} vanishes if a line outage takes place between bus i and bus j.*

6.3 Decentralized Network Inference for Fault Detection and Localization

In light of the computational complexity of fault localization brought by massive scales, a fault detection is performed first. Specifically, let E' be the edge set, $\mathbf{\Sigma}'$ denote the covariance matrix of the GMRF, and $\mathbf{R}'$ be the conditional correlation matrix when the power system is under normal conditions. Mathematically, the proposed fault detection approach boils down to hypothesis testing on the change of the conditional correlation coefficients, with null and alternate hypothesis given by

$$\begin{cases} \mathcal{H}_0 \ : \ r_{ij} = r'_{ij} & \forall \{i,j\} \in E' \\ \mathcal{H}_1 \ : \ r_{ij} \neq r'_{ij} & \exists \{i,j\} \in E' \end{cases} \tag{6.11}$$

It can be shown that the above null hypothesis is equivalent to that of $\mathbf{\Sigma} = \mathbf{\Sigma}'$. Then, it suffices to apply standard change detection methods (see, e.g., [116]). Further, if $\mathcal{H}_1$ is accepted, $|E'|$ locators are deployed to localize all the possible faults:

$$\delta_{ij}\left(d_{ij},\varepsilon\right)=\begin{cases}0 & \text{if } d_{ij}\leqslant\varepsilon \\ 1 & o.w.\end{cases} \qquad \forall\left\{i,j\right\}\in E', \tag{6.12}$$

where, $d_{ij}\triangleq\left|\frac{\hat{r}_{ij}}{r'_{ij}}-1\right|$, $\forall\left\{i,j\right\}\in E'$, are computed using the conditional correlation coefficient $\hat{r}_{ij}$ estimated from PMU measurements, and $\varepsilon>0$ depends on the significant level of fault events.

In performing the above change detection and localization using PMU measurements, we notice a few challenging issues. First, the change of r_{ij} cannot be localized by using the measurements of X_i and X_j only. Note that the measurements of X_i and X_j form a sufficient statistic of the unconditional correlation coefficient ρ_{ij}, which relates to the conditional ones globally, as described by the walk-sum expression [117]:

$$\rho_{ij}=\sum_{l=0}^{\infty}\left(\mathbf{R}^l\right)_{i,j}=\sum_{l=0}^{\infty}\varphi\left(i\overset{l}{\longrightarrow}j\right), \tag{6.13}$$

where, $\varphi\left(i\overset{l}{\longrightarrow}j\right)$ is defined as the walk on G from site i to j with length l. It is clear that ρ_{ij} depends on the products of the conditional correlation coefficients over all the possible walks from i to j, and therefore, the change of ρ_{ij} can be the result of the changes of other entries of $\mathbf{R}$. To tackle this issue, a feasible solution is to obtain a complete estimate of $\mathbf{R}$ from the global measurements. This boils down to obtaining $\hat{\mathbf{J}}$, the estimate of $\mathbf{J}$, from the sample covariance matrix $\hat{\mathbf{\Sigma}}$. However, $\hat{\mathbf{\Sigma}}^{-1}$ cannot be directly used for change localization, since $\hat{\mathbf{\Sigma}}^{-1}$ usually does not have the same sparsity as $\mathbf{J}$, due to noisy measurements or the small number of measurements.

In related work [118], the estimation of the information matrix of a GMRF is often treated as a constrained optimization problem that maximizes the likelihood:

$$\begin{aligned} \mathcal{P}: \text{maximize} \quad & \log\left|\hat{\mathbf{J}}\right| - tr\left(\hat{\mathbf{J}}\hat{\mathbf{\Sigma}}\right) \\ \text{subject to} \quad & \hat{\mathbf{J}}_{ij} = 0, \ \ (i,j) \notin E'. \end{aligned} \tag{6.14}$$

However, solving $\mathcal{P}$ requires centralized computation and global measurements. As noted in [118], the computational complexity could be very high for large-scale problems and current algorithms are not scalable. Also worth mentioning is that the estimation of $\mathbf{J}$ generally requires the number of measurements at least to be comparable to the size of $\mathbf{X}$.

We first devise a decomposition scheme using the multiresolution transform of GMRFs. As noted above, in the traditional multiresolution analysis of GMRFs in related work (e.g., in image processing) where the dependency structure of the original GMRF, usually built on lattice graphs, is irreversibly lost due to transform operations (e.g., subsampling [106] and block-averaging [104]). In contrast, the devised scheme can perfectly reconstruct the original GMRF from the sub-fields. Accordingly, a decentralized network inference algorithm using message passing is designed, to achieve a global solution for fault detection and localization. Compared to the centralized algorithm, the proposed approach has the following salient features. 1) The number of samples, required for the estimation of information matrix using local measurements, is proportional to the size of the largest sub-field other than that of the complete GMRF. 2) The dimension of the subproblems is much smaller, significantly reducing the overall computational complexity. Recall that, in the centralized algorithm, a hypothesis testing for fault detection is employed before the fault localization to mitigate the potential computational complexity incurred in estimating $\mathbf{R}$. In contrast, in the decentralized network inference, we directly perform fault localization based on (6.12). Specifically, for a power system consisting of several subsystems, we decompose the fault localization problem into multiple subproblems, in which the

inference can be carried out based on local measurements. Accordingly, the GMRF is decomposed into corresponding sub-fields, and the sites of the GMRF are classified into *border sites* and *inner sites*, of which the latter are not connected to the other sub-fields. Further, the edges in the original dependency graph are classified into *tie-line edges* which connect different sub-fields, *border-line edges* which connect the border sites of the same sub-field, and *inner-line edges* which have at least one end as inner site. We note that a direct decomposition of the GMRF, by grouping the sites into disjoint subsets, is not capable of capturing the dependency between the sub-fields. With this insight, we construct an additional sub-field by grouping all the border sites in the dependency graph.

In solving the subproblems, a key challenge is that the dependency graphs of the sub-fields are no longer the same as in that of the original GMRF. Indeed, as discussed in [106, 108], the decomposition of MRF would result in a "loss of locality". Consequently, for a GMRF, the information matrices of the sub-fields would have different sparse patterns and non-zero entries from the corresponding blocks of $\mathbf{J}$. Therefore, the knowledge of local information matrix is not sufficient to detect and localize all the faults in the corresponding subproblem.

To tackle the above challenge, we devise a decomposition scheme that can perfectly reconstruct the information matrix of the original GMRF from those of the sub-fields. We first show that for the information matrices of the sub-fields, *the entries corresponding to the inner sites, inner-line edges, and tie-line edges remain the same as in* $\mathbf{J}$ *after the decomposition.* Then, message passing between the sub-fields is employed to "reconstruct" the entries corresponding to the border sites and the border-line edges, that changed due to decomposition.

In what follows, we first study the two-scale decomposition of a GMRF, and demonstrate that $\mathbf{J}$ *can indeed be reconstructed from the information matrices of the*

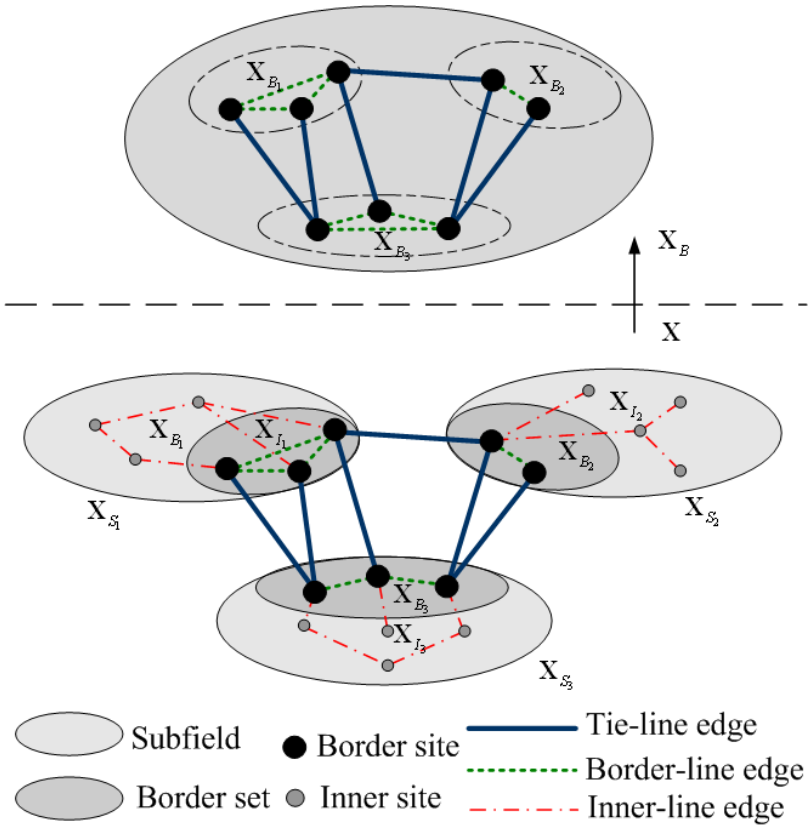


Figure 6.2: The two-scale decomposition of GMRF $\mathbf{X}$

sub-fields through message passing. Then, we extend the study to the multiscale decomposition of GMRFs.

6.3.1 Two-scale Decomposition of GMRF

Simply put, the two-scale decomposition partitions S into K disjoint subsets S_k, $k = 1, 2, \cdots, K$, at the lower scale, and creates one additional subset B at the higher scale by grouping all of the border sites, as shown in Fig. 6.2. For notational convenience, let $\mathbf{X}_{S_k}$ be the sub-field on S_k, $k = 1, 2, \cdots, K$, and $\mathbf{X}_B$ be that on B. Further, we use $\tilde{G}_k$, $\tilde{\mathcal{C}}_k$, and $\mathbf{J}_k$ to denote the dependency graph, the collection of cliques, and the information matrix of $\mathbf{X}_{S_k}$, respectively, for $k = 1, 2, \cdots, K$, and $\tilde{G}_B$, $\tilde{\mathcal{C}}_B$, and $\mathbf{J}_B$ for those of $\mathbf{X}_B$.

Recall that, for a GMRF, the non-zero entries of the information matrix are the coefficients of clique potential functions. By exploring the information matrices of the $K+1$ sub-fields, we find that 1) the off-diagonal entries of $\mathbf{J}_B$ and $\mathbf{J}$ corresponding to the same tie-line edge have the same value; 2) for $\mathbf{J}_k$, $k \in \{1, 2, \cdots, K\}$, and $\mathbf{J}$, the off-diagonal entries corresponding to the same inner-line edge have the same value, and the diagonal entries corresponding to the same inner site have the same value. The above results are summarized in the following two lemmas on the dependency graphs and clique potential functions of the $K+1$ sub-fields, and the proofs are given in Appendix C.

Lemma 6.3.1. ***a****) Let i and j be two border sites contained in different sub-fields, then $i \sim j$ in $\tilde{G}_B$ if and only if $i \sim j$ in G.*
b*) Let c be a clique in $\mathcal{C}$ with a non-zero irreducible potential function. If c contains two sites connected by a tie-line edge, then $c \in \tilde{\mathcal{C}}_B$, and the corresponding potential function is given by $\tilde{V}c(\mathbf{x}_B) = Vc(\mathbf{x}_B, \mathbf{a}_{\bar{B}})$, for any configuration $\mathbf{a}_{\bar{B}} \in \Lambda^{S\setminus B}$ on $\mathbf{X}_{S\setminus B}$.*

Lemma 6.3.2. ***a****) Let i be an inner site, and j be another site in the same sub-field $\mathbf{X}_{S_k}$, $k \in \{1, 2, \cdots, K\}$, then $i \sim j$ in $\tilde{G}_k$ if and only if $i \sim j$ in G.*
b*) Let c be a clique in $\mathcal{C}$ with a non-zero irreducible potential function. If c contains an inner site of S_k, $k \in \{1, 2, \cdots, K\}$, then $c \in \tilde{\mathcal{C}}_k$, and the corresponding potential function is given by $\tilde{V}c(\mathbf{x}_{S_k}) = Vc\left(\mathbf{x}_{S_k}, \mathbf{a}_{\bar{S}_k}\right)$, for any configuration $\mathbf{a}_{\bar{S}_k} \in \Lambda^{S\setminus S_k}$ on $\mathbf{X}_{S\setminus S_k}$.*

Further, in order to enable the complete reconstruction of $\mathbf{J}$, we consider the border sites, the border-line edges, and the corresponding potential functions. For notational convenience, let B_k and I_k be the set of the border sites and inner sites in S_k, respectively, and $\mathbf{X}_{B_k}$ and $\mathbf{X}_{I_k}$ the corresponding random fields, for $k =$

$1, 2, \cdots, K$. Note that $B = \cup_{k=1}^{K} B_k$. Without loss of generality, GMRF $\mathbf{X}$ could be organized as

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_B \\ \mathbf{X}_I \end{bmatrix},$$

with

$$\mathbf{X}_B = \begin{bmatrix} \mathbf{X}_{B_1} \\ \vdots \\ \mathbf{X}_{B_K} \end{bmatrix} \quad \mathbf{X}_I = \begin{bmatrix} \mathbf{X}_{I_1} \\ \vdots \\ \mathbf{X}_{I_K} \end{bmatrix}.$$

Then, $\mathbf{J}$ consists of four block matrices $\mathbf{J}_{BB}$, $\mathbf{J}_{BI}$, $\mathbf{J}_{IB}$, and $\mathbf{J}_{II}$, with the border-line edges and border sites corresponding to the diagonal blocks of $\mathbf{J}_{BB}$. According to Lemma 6.3.2, the non-zero entries of $\mathbf{J}_{BI}$, $\mathbf{J}_{IB}$, and $\mathbf{J}_{II}$ have the same value as those of $\mathbf{J}_k,\ k = 1, 2, \cdots, K$ corresponding to the same inner-line edge or inner site. Another key observation on the block matrices of $\mathbf{J}$ and $\mathbf{J}_B$ is given by the following result.

Proposition 6.3.1. *The k-th diagonal blocks of* $\mathbf{J_{BB}}$ *is given by*

$$[\mathbf{J}_{BB}]_{kk} = [\mathbf{J}_B]_{kk} + [\mathbf{J}_{BI}]_{kk} [\mathbf{J}_{II}]_{kk}^{-1} [\mathbf{J}_{BI}]_{kk}^{T}, \tag{6.15}$$

where, $[\ \cdot\ ]_{kk}$ *retrieves the k-th diagonal block of an information matrix.*

Proposition 6.3.1 indicates that the entries of $\mathbf{J}$ corresponding to the border-line edges and border sites could be computed from $\mathbf{J}_B$, and the entries of $\mathbf{J}$ that corresponds to the inner-line edges and inner sites. Combining Lemma 6.3.1, Lemma 6.3.2 with Proposition 6.3.1, we conclude that $\mathbf{J}$ could be completely reconstructed from $\mathbf{J}_B$ and $\mathbf{J}_k,\ k = 1, 2, \cdots, K$, through message passing between the sub-fields. For large-scale GMRFs, even if the size of each sub-field $\mathbf{X}_{S_k}$ is moderate, the size of $\mathbf{X}_B$ could still be very large, which makes the estimation of $\mathbf{J}_B$ intractable. This problem could be solved by applying further decomposition on $\mathbf{X}_B$.

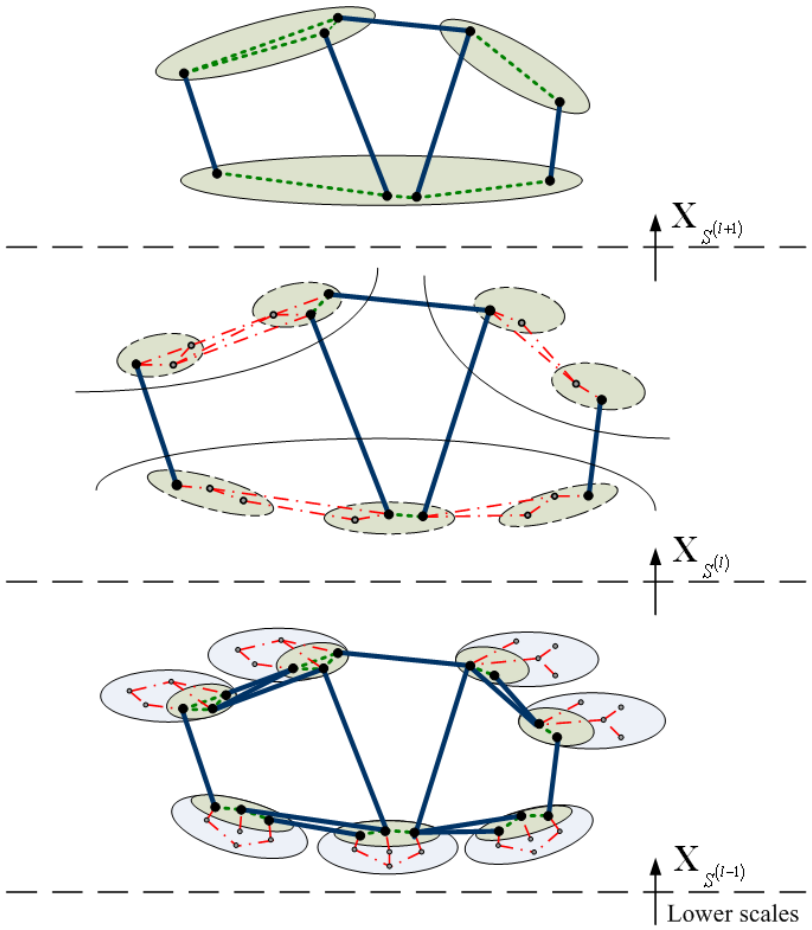


Figure 6.3: The multiscale decomposition of GMRF $\mathbf{X}$

6.3.2 Multiscale Decomposition of GMRF

According to Lemma 6.3.1, one characteristic of the proposed two-scale decomposition of GMRFs is that the tie-line structure of G is preserved in $\tilde{G}_B$. Accordingly, the two-scale decomposition could be applied to $\mathbf{X}_B$, the newly created sub-field at the higher scale, recursively as needed.

In general, for the L-scale decomposition, we first perform a hierarchical clustering on S, as illustrated in Fig. 6.3. For notational convenience, we use the superscript as the index of scale. Specifically, $S^{(2)}$ is used to denote the set of sites at the

second scale, instead of B in the two-scale decomposition scenario. Suppose that $S^{(l)}$ is obtained by grouping the border sites of the $K^{(l-1)}$ disjoint subsets of $S^{(l-1)}$. To obtain $S^{(l+1)}$, we re-group the scale-l sites into $K^{(l)}$ disjoint subsets of $S^{(l)}$, such that each pair of sites in the same subset of $S^{(l-1)}$ are still contained in the same subset of $S^{(l)}$. In each scale-l subset $S_k^{(l)}$, the sites and edges are re-classified according to the same rule as in the two-scale decomposition, and then $S^{(l+1)}$ could be obtained by grouping all the border sites of $S^{(l)}$. As a result, the entire set of sites S is partitioned into the disjoint subsets of scale-1 inner sites, ..., scale-$(L-1)$ inner sites, and scale-$(L-1)$ border sites, and the entire set of edges E into those of the scale-1 inner-line edges, ..., scale-$(L-1)$ inner-line edges, scale-$(L-1)$ border-line edges, and scale-$(L-1)$ tie-line edges which are also scale-L edges since we cease clustering at the scale L.

Summarizing, once the information matrices $\mathbf{J}_k^{(l)}$ of the sub-fields $\mathbf{X}_{S_k^{(l)}}$, $k = 1, 2, \cdots, K^{(l)}$, across all L scales are known, the complete $\mathbf{J}$ could be obtained by performing the reconstruction procedure recursively from the top to the lowest scale, using Proposition 6.3.1, for the reconstruction at each scale. With this insight, we design a decentralized network inference algorithm via using message passing for the estimation of the information matrix.

6.3.3 Decentralized Network Inference using Message Passing

Simply put, if all the buses of the power system are observable, we first perform a multiscale decomposition on $\mathbf{X}$ based on the hierarchical topology of the power system. Once the estimates of the information matrices of sub-fields are obtained, a complete $\hat{\mathbf{J}}$ could be reconstructed from the estimated information matrices of sub-fields. For each scale l, $l = 1, 2, \cdots, L$, we assume that there is an inference center at each sub-field $\mathbf{X}_{S_k^{(l)}}$, $k = 1, 2, \cdots, K^{(l)}$. Let $F(k, l)$ be the collection of the indices of

the sub-fields that are located at the lower scale of $\mathbf{X}_{S_k^{(l+1)}}$. Then, the procedure of the decentralized estimation of the information matrix is summarized in Algorithm 5.

Algorithm 5 Decentralized estimation of $\mathbf{J}$ using message passing

Local estimation: Estimate the information matrices of all the sub-fields based on local measurements, by solving the subproblem $\mathcal{P}_k^l$ using the dependency graph of $\mathbf{X}_{S_k^{(l)}}$.

Down-top message passing: For $l = 1, 2, \cdots, L-1$, the inference centers of $\mathbf{X}_{S_f^{(l)}}$, $f \in F(k,l)$, submit $\hat{\mathbf{J}}_f^{(l)}$ to that of $\mathbf{X}_{S_k^{(l+1)}}$.

Top-down reconstruction: For $l = L-1, L-2, \cdots, 1$, the inference center of $\mathbf{X}_{S^{(L)}}$ reconstruct $\hat{\mathbf{J}}^{(l)}$ from $\hat{\mathbf{J}}^{(l+1)}$ and $\hat{\mathbf{J}}_k^{(l)}$, $k = 1, 2, \cdots, K^{(l)}$.

Top-down message passing: The inference center of $\mathbf{X}_{S^{(L)}}$ broadcast $\hat{\mathbf{J}}$, i.e., $\hat{\mathbf{J}}^{(1)}$ to the inference centers of all the sub-fields.

We note that in some practical scenarios, the power system is not completely observable, e.g., due to the failures of some PMUs. As a result, the complete $\hat{\mathbf{J}}$ is not obtainable. However, fault diagnosis could still be performed for a sub-field under mild conditions. To this end, we have the following result.

Corollary 6.3.1. *The measurements at a sub-field $\mathbf{X}_{S_k}$ and its neighbor sites form a sufficient statistic for $\mathbf{J}_{kk}$.*

The above result indicates that the information matrix of the interior of an observable island can be estimated from the measurements of the island only. To get a more concrete sense, suppose sub-field $\mathbf{X}_{\mathbf{S_3}}$ and its neighbor sites are observable in Fig. 6.2. We first construct an extended sub-field $\bar{\mathbf{X}}_{\mathbf{S_3}}$ as the concatenation of $\mathbf{X}_{\mathbf{S_3}}$ and its neighbor sites, as shown in Fig. 6.4. Then, according to Corollary 6.3.1, $\mathbf{J}_{33}$ is same as the block of the information matrix of $\bar{\mathbf{X}}_{\mathbf{S_3}}$, that corresponds to $\mathbf{X}_{\mathbf{S_3}}$.

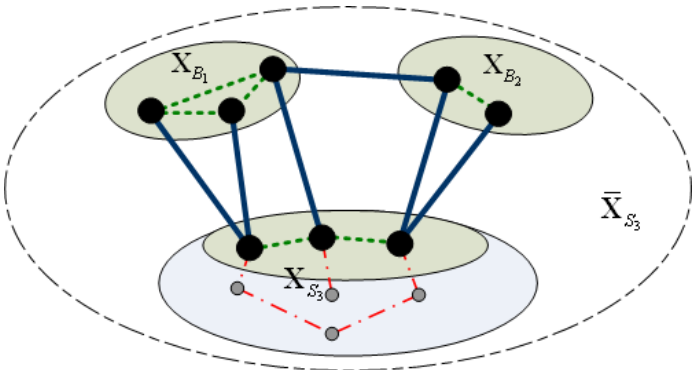


Figure 6.4: The extended sub-field for the fault diagnosis of S_3

6.4 Numerical Results

6.4.1 Decentralized Estimation of Information Matrix

In what follows, we perform numerical experiments to evaluate the proposed decentralized estimation algorithm. Following the approach in [119], we consider a GMRF with $p = 300$ sites uniformly distributed on a 40×40 grid. Any two sites i, j are considered to be neighbors whenever their Euclidean distance $d(i, j)$ is less than a cut-off value d_0. Further, the conditional correlation coefficients of adjacent sites are parameterized as $r_{ij} = \frac{\gamma}{d^2(i,j)}$. We choose $\gamma \in [0, 1]$ such that $(\mathbf{I} - \mathbf{R})$ is positive definite. We generate $i.i.d$ samples and perform the estimation procedure. For the multiscale estimation, we apply the clustering algorithm in [120] and group the GMRF into 16 clusters at the second scale, and 4 clusters at the third scale. The accuracy of estimation is quantified by the Kullback-Leibler (KL) distance between the true distribution $\mathcal{N}_0(0, \mathbf{J}^{-1})$ and the distribution $\mathcal{N}_1(0, \hat{\mathbf{J}}^{-1})$ that is estimated from measurements:

$$D\left(\mathcal{N}_0||\mathcal{N}_1\right) = \frac{1}{2}\left(\log\det\left(\mathbf{J}\hat{\mathbf{J}}^{-1}\right) + tr\left(\mathbf{J}^{-1}\hat{\mathbf{J}}\right)\right). \tag{6.16}$$

We vary the values of d_0 and γ to generate various simulation cases with different graph density and strength of conditional correlations. As shown in Fig. 6.5 and

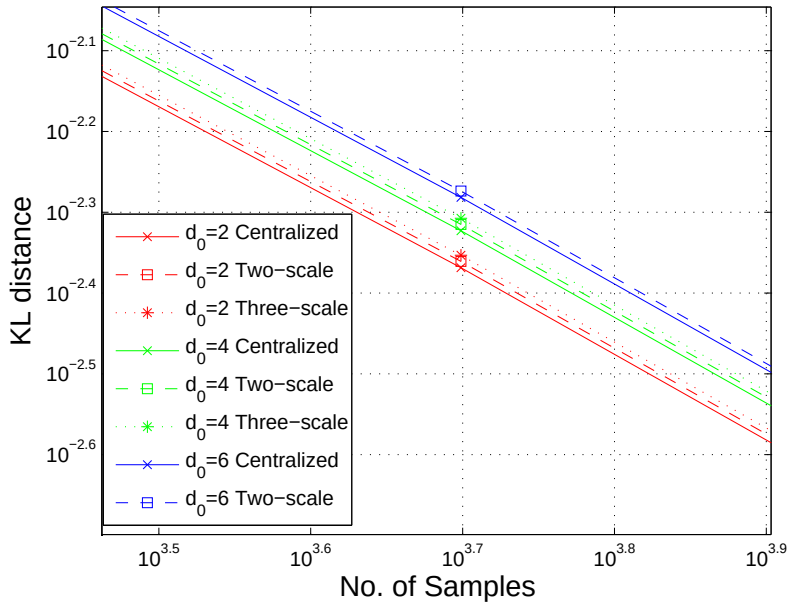


Figure 6.5: The performance of estimation algorithms ($\gamma = 0.5$)

Fig.6.6, the numerical results indicate that the KL distances decrease approximately in a power law against the number of samples, and the proposed decentralized algorithms have comparable accuracy with the centralized one. For $d = 6$, graph for the sub-field at the 2nd scale is so dense that further clustering gives no gains. They also indicate that the accuracy of the estimation algorithms does not depend on the topology of graph, or the strength of conditional correlations.

6.4.2 Fault Localization

In order to evaluate the effectiveness of the proposed approach, we run simulations on the IEEE 300-bus system. Following the Monte Carlo methods, we consider the fault detection and localization in a 20-second observation window, i.e., $n = 1200$ samples are simulated since phasors are measured at up to 60 samples per second according to the specifications of PMU. Power injections to buses are generated randomly, and then the branch flows and phasors at buses are solved using the MATPOWER [121]

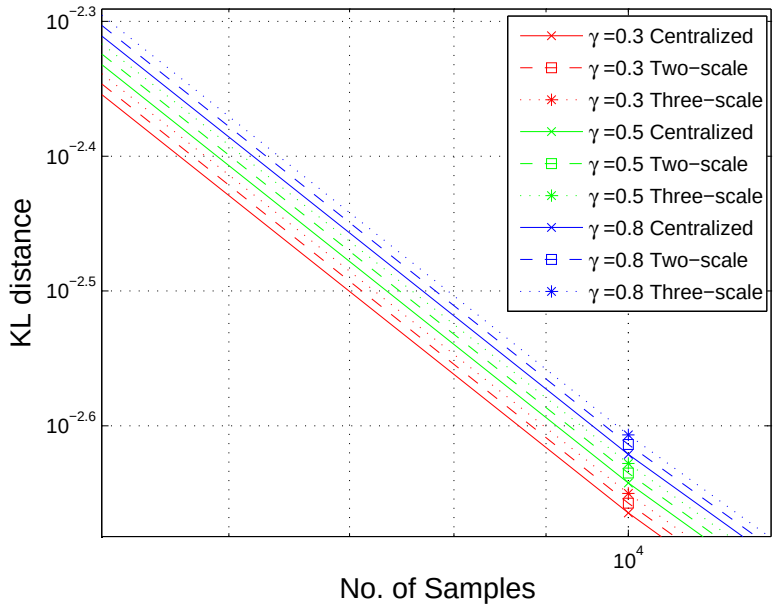


Figure 6.6: The performance of estimation algorithms ($d_0 = 4$)

simulation package. We take the solution of the phasors as the measurements collected from the IEEE 300-bus system by PMUs .

Based on the locators defined in (6.12), we specify the detection rules for the two categories of fault events: line outage and the change of physical parameter. In the event of a line outage between the buses i and j, X_i and X_j become conditionally un-correlated, and then, we could use the following rule for the identification of line outage: *Line outage at* $\{i, j\}$, *if* $r'_{ij} \gg \epsilon$ *and* $\hat{r}_{ij} < \epsilon$, where $\epsilon \ll 1$ is a small value. For the fault event of a parameter change (i.e., caused by physical damages), the increase of the electrical distance between buses would result in less conditionally-correlated phase angles. Therefore, we could use the following rule for identifying these faults: *Line* $\{i, j\}$ *with parameter change, if* $r'_{ij}/\hat{r}_{ij} \geqslant \eta$, where $\eta > 1$ depends on the significant level of the fault events to be detected.

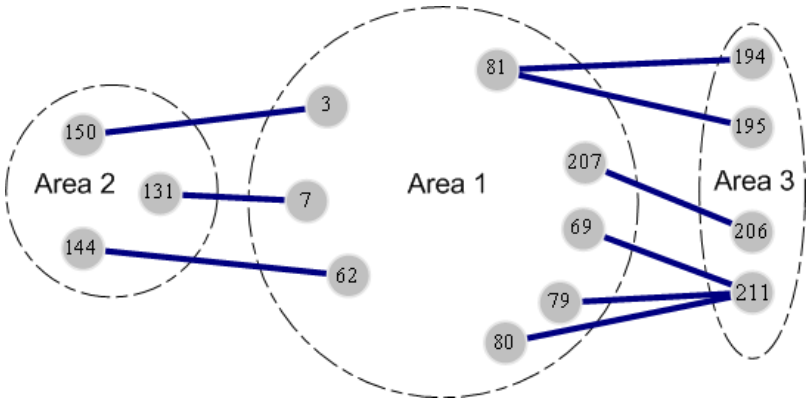


Figure 6.7: The top-scale sub-field of the IEEE 300-bus System

6.4.2.1 Decomposition of the IEEE 300-bus System

We perform a two-scale decomposition on the graph of the phase angles in the IEEE 300-bus System, as shown in Fig. 6.7. Since the IEEE 300-bus system contains three management areas, we take the three areas as the sub-fields at the lower scale, and the top-scale sub-field is obtained by grouping all the border sites of the sub-fields at the lower scale. It is worth noting that only the tie-line edges are shown in Fig. 6.7, and the border sites in each sub-field form a complete subgraph.

6.4.2.2 Testing cases for line outage

We test a case of line {26,27} outage. To simulate the line outage, we exclude the failed line from solving the power flows. Once an estimate of the information matrix of the GMRF are obtained from the measurements of phase angles, the conditional correlation coefficients are computed, as shown in Table 6.1. It indicates that the conditional correlation coefficient between the phaosr angles at the buses 26 and 27 turns into a small value. Thus a decision rule with $\epsilon = 0.05$ could detect this line outage.

Table 6.1: A case of line {26,27} outage

	line	Area	Reactance	Con. Cor. Coef.	
				$\hat{r}'$	$\hat{r}$
Bus 26	{25,26}	1	0.071	0.61	0.62
	{26,27}	1	0.12	0.36	0.03
	{26,320}	1	0.13	0.34	0.39
Bus 27	{20,27}	1	0.186	0.23	0.25
	{26,27}	1	0.12	0.36	0.03

6.4.2.3 Testing cases for the change of physical parameter

We test a case that both line {26,27} and line {199,200} have parameter change. In this case, we double the reactance of the faulted transmission lines to simulate an increase of electrical distance. The results are summarized in Table 6.2. Note that if we choose $\eta = 1.3$, then the faulted lines are detected.

Table 6.2: A case of parameter changes at line {26,27} and {199,200}

	line	Area	Reactance		Con. Cor. Coef.	
			$1/b'_{ij}$	$1/b_{ij}$	r'	r
Bus 26	{25,26}	1	0.071	0.071	0.61	0.72
	{26,27}	1	0.12	0.24	0.36	0.21
	{26,320}	1	0.13	0.13	0.34	0.39
Bus 27	{20,27}	1	0.186	0.186	0.23	0.43
	{26,27}	1	0.12	0.24	0.36	0.21
Bus 199	{199,200}	2	0.135	0.27	0.53	0.38
	{199,210}	2	0.102	0.102	0.68	0.75
Bus 200	{199,200}	2	0.135	0.27	0.53	0.38
	{200,210}	2	0.128	0.128	0.58	0.63
	{200,248}	2	0.22	0.22	0.38	0.41

6.5 Conclusion

A dependency graph approach is proposed for the fault detection and localization of transmission lines in large scale power systems. Then, the fault diagnosis is performed through the change detection and localization in the conditional correlation matrix

of the GMRF. In particular, in order to mitigate the complexity, a decentralized algorithm based on multiresolution transform of GMRFs is devised. The proposed decentralized algorithms can be useful in some practical scenarios, e.g, when the PMU measurements from multiple utilities are incompatible (not synchronized or at different sampling frequencies).

Chapter 7

Future Research

7.1 Impact of Communication QoS on the Performance of Special Protection Schemes

Special protection schemes (SPS), also known as remedial action schemes (RAS), are typically contingency-related schemes that are designed to initiate pre-planned, corrective action upon disturbance and abnormal operating conditions are detected [122]. SPS actions may include load shedding, generator tripping, or change in system configuration (e.g., shunt banks inserting, capacitor bypassing) to maintain system frequency stability and acceptable voltages levels. The trigger of SPS is usually a logical combination of the status signals that are collected from multiple remote substations.

With this insight, it is obvious that the overlay communication system can have significant impact on the performance of SPS. First, the reliability of the communication system determines the availability of these input signals. Further, large latency incurred by wide-area communication may lead the failure of SPS to meet the timing requirement of correction actions. Therefore, it is necessary to perform reliability analysis and risk assessment of SPS under a two-layer framework, i.e., a power system with an overlay communication system.

Basically, the communication system for wide-area protection system can be divided into two parts: intra-substation communication and inter-substation communication. The task of intra-substation communication network, i.e., a local area network (LAN), is to collect the status data of lines and switchgears and report it to the server within the substation; and inter-substation communications aim to deliver the needed status data in a timely manner to the control center or the site of

SPS. Inter-substation communications can be implemented through dedicated point-to-point communication technologies, e.g., microwave and optical-fiber, or through a wide-area network (WAN). Reliability and latency analysis will be carried out for both parts of the communication system, and for all possible implementation scenarios.

7.2 Near Real-time Prediction and Quantification of Extreme Ramp-down Events of Wind Farm Generation

Ramp events of wind farm generation refer to the rapid change in the farm's aggregate power output [123]. Ramp is usually characterized by the magnitude of change and the length of the considered time interval. Extreme ramp-down events, e.g., a reduction of 20% of the rated capacity in 10 minutes, can have significant impact on the efficiency of wind generation integration and power system reliability if ancillary services are not sufficient. This kind of low probability high impact events have to be taken into account in wind farm generation forecast and power system operations.

A vast majority of existing forecasting methods, which are built on time-series analysis or Markovian models, may not be appropriate to handle extreme ramp-down event Time-series analysis approaches, e.g., ARMA model based approaches, implicitly assume that wind farm generation is a stationary or quasi-stationary process, thus cannot capture the non-stationarity of the rapid change of wind farm generation when extreme ramp-down events occur. Markovian models, e.g., Markov chains, may also fail when extreme ramp-down events are under-represented in the historic data that is used to estimated the transition probabilities.

Thus motivated, near real-time prediction of extreme ramp-down events and quantification of ramp magnitude will be studied in a new framework. In this framework, individual turbine's power output, combined with neighborhood graph obtained by using the geographical information, will be utilized to predict the occurrence of ex-

treme ramp-down events, by using tools from graphical learning and inference. From the extensive measurement data, two major causes of extreme ramp-down events are identified: steep wind speed drop, and diversion of wind direction combined with speed drop. For the latter case, real-time detection of wind direction change is more challenging, since there is limited number of MET towers even for large wind farms. Further, when quantifying the ramp magnitude, wake effect needs to be accounted for when predicting individual turbine's power output.

7.3 Data Security and Privacy in Smart Grids

Besides reliability of power systems, another key objective of smart grids is to improve the efficiency of the electricity grid, by using ICTs that enabling two-way communications between end-users and utilities. It is thus obvious that data containing detailed usage information of end users will be collected by utilities and/or third parties; the utility can provide real-time pricing and billing information to smart meters and other end-user devices. From the perspective of utilities, AMI and the overlay information system will allow for a more flexible monitoring and control on the distribution grid.

Undoubtedly, incorporating smartness to a critical electricity distribution grid imposes stringent requirements on data security and privacy. A security breach of the information system of distribution management system can have severe impact on the reliability of distribution grid. With regards to privacy, the metering data collected by the utility contains a significant amount of information about individual end-users. The potential implications of data privacy have already been identified as another most important issue of smart grid, besides interoperability, by National Institute of Standards and Technology (NIST) [124].

It is thus urgent to deploy data security and privacy protection and assurance schemes into the overlay information infrastructure. Specifically, the following

aspects needs to be taken into account. 1) Communication security: two-way information exchange will require new ways of secure communications between end-users and utilities, as well as new ways to build secure communication networks within the home area. 2) Implementation security: With a huge number of small devices embedded in electricity appliances are getting networked, implementation security issues become significant. Conventionally, those devices are not equipped with protection schemes against network-based attacks, and the limited hardware and software resources of these devices make implementation of security solutions what are designed for larger computer systems impossible. Thus, new implementation security schemes are required. 3) Privacy: The amount of smart metering data collected from individual end-users is unprecedented, and leads to massive concerns about users privacy. Flexible privacy-conserving technologies are required to balance the tradeoff between end-user privacy and data usage. 4) Grid Architectures: The smart grid combines architectural requirements that are inherently contradictory. On one side, grid protection and control schemes always put more wights on the availability of data. On the other side, this particular privacy related and security critical data require adequate protection schemes. New architectures need to be designed to accommodate both privacy and dependability.

REFERENCES

[1] S.Grijalva, "Computational challenges and analysis under increasingly dynamic and uncertain electric power system conditions ," in *Power and Energy Society General Meeting, 2012 IEEE*, July 2012, pp. 1–2.

[2] CERTS, "Phasor technology advanced concepts FAQ," [Online] Available: `http://www.phasor-rtdms.com/phaserconcepts/phasor_adv_faq.html`.

[3] "Energy Independence and Security Act of 2007 [Public Law No: 110-140] Title XIII, Sec. 1301.," [Online] Available: `http://www.gpo.gov/fdsys/pkg/BILLS-110hr6enr/pdf/BILLS-110hr6enr.pdf`.

[4] NERC, "Reliability standards: Glossary of terms," [Online] Available: `http://www.nerc.com/page.php?cid=2|20`.

[5] "Accommodating high levels of variable generation," NERC Special Report, [Online] Available: `www.nerc.com/files/IVGTF_Report_041609.pdf`.

[6] D. Lew, M. Milligan, G. Jordan, and R. Piwko, "The value of wind power forecasting," *NREL Conference Paper CP-5500-50814*, Apr. 2011.

[7] K. S. Cory and B. G. Swezey, "Renewable portfolio standards in the states: balancing goals and implementation strategies," *NREL Technical Report TP-670-41409*, Dec. 2007.

[8] E. A. DeMeo, G. A. Jordan, C. Kalich, J. King, M. R. Milligan, C. Murley, B. Oakleaf, and M. J. Schuerger, "Accommodating wind's natural behavior," *IEEE Power Energy Mag.*, vol. 5, no. 6, pp. 59–67, Nov.-Dec. 2007.

[9] S. Fink, C. Mudd, K. Porter, and B. Morgenstern, "Wind energy curtailment case studies," NREL Subcontract Report SR-550-46716, Oct. 2009.

[10] P. Pinson and H. Madsen, "Adaptive modelling and forecasting of offshore wind power fluctuations with Markov-switching autoregressive models," *Journal of Forecasting*, vol. 31, no. 4, pp. 281–313, 2012.

[11] F. Cassola and M. Burlando, "Wind speed and wind energy forecast through Kalman filtering of numerical weather prediction model output," *Applied Energy*, vol. 99, pp. 154–166, 2012.

[12] G. Papaefthymiou and B. Klockl, "MCMC for wind power simulation," *IEEE Trans. on Energy Convers.*, vol. 23, no. 1, pp. 234–240, Mar. 2008.

[13] A. Carpinone, R. Langella, A. Testa, and M. Giorgio, "Very short-term probabilistic wind power forecasting based on Markov chain models," in *Probabilistic Methods Applied to Power Systems (PMAPS), 2010 IEEE 11th International Conference on*, June 2010, pp. 107–112.

[14] S. Santoso, M. Negnevitsky, and N. Hatziargyriou, "Data mining and analysis techniques in wind power system applications: abridged," in *Power Engineering Society General Meeting, 2006. IEEE*, 2006, pp. 1–3.

[15] A. Kusiak, H. Zheng, and Z. Song, "Wind farm power prediction: a data-mining approach," *Wind Energy*, vol. 12, no. 3, pp. 275–293, 2009.

[16] G. Giebel, R. Brownsword, G. Kariniotakis, M. Denhard, and C. Draxl, *The State of the Art in Short-Term Prediction of Wind Power - A Literature Overview*. ANEMOS.plus, 2011, [Online] Available: `http://www.anemos-plus.eu/images/pubs/deliverables/aplus.deliverable_d1%.2.stp_sota_v1.1.pdf`.

[17] C. Monteiro, H. Keko, R. Bessa, V. Miranda, A. Botterud, J. Wang, and G. Conzelmann, "A quick guide to wind power forecating: state-of-the-art 2009," [Online] Available: `http://www.dis.anl.gov/pubs/65614.pdf`, 2009.

[18] A. Lau and P. McSharry, "Approaches for multi-step density forecasts with application to aggregated wind power," *Ann. Appl. Stat.*, vol. 4, no. 3, pp. 1311–1341, 2010.

[19] P. Pinson, "Very-short-term probabilistic forecasting of wind power with generalized logit-normal distributions," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 61, no. 4, pp. 555–576, 2012.

[20] M. He, L. Yang, J. Zhang, and V. Vittal, "Spatio-temporal analysis for smart grids with wind generation integration," in *Computing, Networking and Communications (ICNC), 2013 International Conference on*, 2013, pp. 1107–1111.

[21] P. Luh, Y. Yu, B. Zhang, E. Litvinov, T. Zheng, F. Zhao, J. Zhao, and C. Wang, "Grid Integration of Intermittent Wind Generation: a Markovian Approach," in press, *IEEE Trans. Smart Grids*.

[22] M. E. J. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemporary Physics*, vol. 46, no. 5, pp. 323–351, 2005.

[23] G. Samorodnitsky and M. Taqqu, *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance (Stochastic Modeling Series)*. Chapman and Hall/CRC, 1994.

[24] E. S. Tackle and J. M. Brown, "Note on the use of Weibull statistics to characterize wind speed data," *Journal Appl. Meteorol*, vol. 17, pp. 556–559, 1978.

[25] G. C. Stone and G. V. Heeswijk, "Parameter estimation for the Weibull distribution," *IEEE Trans. on Elect. Insul.*, vol. EI-12, no. 4, pp. 253 – 261, Aug. 1977.

[26] N. Abdel-Karim, M. Small, and M. Ilic, "Short term wind speed prediction by finite and infinite impulse response filters: A state space model representation using discrete Markov process," *IEEE Bucharest Power Tech Conference*, 2009.

[27] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[28] Q. Zhang and S. A. Kassam, "Finite-state Markov model for Rayleigh fading channels," *IEEE Trans. on Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov. 1999.

[29] D. Kugiuntzis and E. Bora-Senta, "Gaussian analysis of non-Gaussian time series," *Brussels Economic Review*, vol. 53, no. 2, pp. 295–322, 2010.

[30] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control, 2nd ed.* San Francisco: Holden-Day, 1976.

[31] NERC IVGTF Task 2.1 report: Variable Generation Power Forecasting for Operations, [Online] Available: `www.nerc.com/docs/pc/ivgtf/Task2-1(5.20).pdf`, May 2010.

[32] H. Madsen, P. Pinson, G. Kariniotakis, H. A. Nielsen, and T. S. Nielsen, "Standardizing the performance evaluation of short-term wind power prediction models," *Wind Engineering*, vol. 29, no. 6, pp. 475–489, 2005.

[33] L. Xie, P. Carvalho, L. Ferreira, J. Liu, B. Krogh, N. Popli, and M. Ilic, "Wind integration in power systems: operational challenges and possible solutions," *Proc. IEEE*, vol. 99, Jan 2011.

[34] P. Varaiya, F. Wu, and J. Bialek, "Smart operation of smart grid: Risk-limiting dispatch," *Proc. IEEE*, vol. 99, no. 1, pp. 40–57, Jan 2011.

[35] L. Xie and M. Ilic, "Model predictive economic/environmental dispatch of power systems with intermittent resources," in *IEEE PES General Meeting*, July 2009, pp. 1–6.

[36] F. Bouffard and F. Galiana, "Stochastic security for operations planning with significant wind power generation," *IEEE Trans. Power Syst.*, vol. 23, no. 2, pp. 306–316, May 2008.

[37] M. Ortega-Vazquez and D. Kirschen, "Estimating the spinning reserve requirements in systems with significant wind power generation penetration," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 114–124, Feb 2009.

[38] J. Morales, A. Conejo, and J. Perez-Ruiz, "Economic valuation of reserves in power systems with high penetration of wind power," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 900–910, May 2009.

[39] S. Newman, "Smart appliances for an energy-efficient future," [Online] Available: `http://green.yahoo.com`, 2008.

[40] R. Sioshansi, "Evaluating the impacts of real-time pricing on the cost and value of wind generation," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 741 –748, may 2010.

[41] J. Bialek, P. Varaiya, F. Wu, and J. Zhong, "Risk-limiting dispatch of smart grid," in *IEEE PES General Meeting*, July 2010, pp. 1–2.

[42] S. E. Fleten and E. Pettersen, "Constructing bidding curves for a price-taking retailer in the norwegian electricity market," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 701–708, May 2005.

[43] S. Borenstein, "Customer risk from real-time retail electricity pricing: Bill volatility and hedgability," *Energy Journal*, vol. 28, no. 2, pp. 111–130, 2007.

[44] K. Cory and B. Swezey, "Renewable portfolio standards in the states: Balancing goals and implementation strategies," NREL Technical Report TP-670-41409, Dec. 2007.

[45] M. Mahmoud, "Multilevel systems control and applications," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, no. 3, pp. 125–143, Jun 1977.

[46] H. S. Chang, P. J. Fard, S. I. Marcus, and M. Shayman, "Multitime scale markov decision processes," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 976–987, Jun 2003.

[47] E. A. Feinberg and A. Shwartz, *Handbook of Markov Decision Processes*. Springer Netherlands, Jun 2005.

[48] K. Hedman, M. Ferris, R. O'Neill, E. Fisher, and S. Oren, "Co-optimization of generation unit commitment and transmission switching with n-1 reliability," *IEEE Trans. on Power Syst.*, vol. 25, no. 2, pp. 1052–1063, May 2010.

[49] C. Grigg *et al.*, "The IEEE reliability test system-1996," *IEEE Trans. on Power Syst.*, vol. 14, no. 3, pp. 1010–1020, Aug 1999.

[50] E. Ela, M. Milligan, and B. Kirby, "Operating reserves and variable generation," NREL Technical Report TP-5500-51978, Aug. 2011.

[51] P. Sauer, K. L. Tomsovic, and V. Vittal, *Dynamic Security Assessment*, 2nd ed., ser. The Electric Power Engineering Handbook. CRC Press, 2007, chapter 15, pp. 1–10.

[52] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for online transient stability assessment of power systems," *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 789–800, May 1989.

[53] L. Wehenkel, M. Pavella, E. Euxibie, and B. Heilbronn, "Decision tree based transient stability method: a case study," *IEEE Trans. Power Syst.*, vol. 9, no. 1, pp. 459–469, feb 1994.

[54] S. Rovnyak, S. Kretsinger, J. Thorp, and D. Brown, "Decision trees for real-time transient stability prediction," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1417–1426, Aug 1994.

[55] K. Sun, S. Likhate, V. Vittal, V. Kolluri, and S. Mandal, "An online dynamic security assessment scheme using phasor measurements and decision trees," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1935–1943, Nov 2007.

[56] R. Diao, K. Sun, V. Vittal, R. O'Keefe, M. Richardson, N. Bhatt, D. Stradford, and S. Sarawgi, "Decision tree-based online voltage security assessment using PMU measurements," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 832–839, May 2009.

[57] R. Diao, V. Vittal, and N. Logic, "Design of a real-time security assessment tool for situational awareness enhancement in modern power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 957–965, May 2010.

[58] V. Miranda, J. Fidalgo, J. Lopes, and L. Almeida, "Real time preventive actions for transient stability enhancement with a hybrid neural network-optimization approach," *IEEE Trans. Power Syst.*, vol. 10, no. 2, pp. 1029–1035, May 1995.

[59] C. Jensen, M. El-Sharkawi, and R. Marks, "Power system security assessment using neural networks: feature selection using Fisher discrimination," *IEEE Trans. Power Syst.*, vol. 16, no. 4, pp. 757–763, Nov. 2001.

[60] I. Kamwa, R. Grondin, and L. Loud, "Time-varying contingency screening for dynamic security assessment using intelligent-systems techniques," *IEEE Trans. Power Syst.*, vol. 16, no. 3, pp. 526–536, Aug. 2001.

[61] L. Moulin, A. da Silva, M. El-Sharkawi, and I. Marks, R.J., "Support vector machines for transient stability analysis of large-scale power systems," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 818–825, May 2004.

[62] A. Rajapakse, F. Gomez, K. Nanayakkara, P. Crossley, and V. Terzija, "Rotor angle instability prediction using post-disturbance voltage trajectories," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 947–956, May 2010.

[63] F. Gomez, A. Rajapakse, U. Annakkage, and I. Fernando, "Support vector machine-based algorithm for post-fault transient stability status prediction using synchronized measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1474–1483, Aug. 2011.

[64] I. Kamwa, S. Samantaray, and G. Joos, "Development of rule-based classifiers for rapid stability assessment of wide-area post-disturbance records," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 258–270, Feb. 2009.

[65] I. Kamwa, "Catastrophe predictors from ensemble decision-tree learning of wide-area severity indices," *IEEE Transactions on Smart Grid*, vol. 1, no. 2, pp. 144–158, Sep. 2010.

[66] M. He, J. Zhang, and V. Vittal, "A data mining framework for online dynamic security assessment: decision trees, boosting, and complexity analysis," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, Jan. 2012, pp. 1–8.

[67] 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 Trans. Power Syst.*, vol. 27, no. 3, pp. 1253–1263, Aug. 2012.

[68] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, ser. Springer Series in Statistics. Springer-Verlag, 2008.

[69] I. Genc, R. Diao, V. Vittal, S. Kolluri, and S. Mandal, "Decision tree-based preventive and corrective control applications for dynamic security enhancement in power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1611–1619, Aug. 2010.

[70] I. Kamwa, S. Samantaray, and G. Joos, "On the accuracy versus transparency trade-off of data-mining models for fast-response PMU-based catastrophe predictors," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 152–161, March 2012.

[71] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Chapman and Hall/CRC, 1984.

[72] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[73] A. Niculescu-mizil and R. Caruana, "Obtaining calibrated probabilities from boosting," in *In: Proc. 21st Conference on Uncertainty in Artificial Intelligence (UAI 05), AUAI Press*. AUAI Press, 2005.

[74] R. Banfield, L. Hall, K. Bowyer, and W. Kegelmeyer, "A comparison of decision tree ensemble creation techniques," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 173–180, Jan. 2007.

[75] P. Utgoff, N. Berkman, and J. Clouse, "Decision tree induction based on efficient tree restructuring," *Mach. Learn.*, vol. 29, pp. 5–44, Oct 1997.

[76] M. Box, D. Davies, and W. Swann, *Non-Linear optimisation Techniques*. Oliver and Boyd, 1969.

[77] R. Schainker, G. Zhang, P. Hirsch, and C. Jing, "On-line dynamic stability analysis using distributed computing," in *Power and Energy Society General Meeting, 2008 IEEE*, July 2008, pp. 1–7.

[78] M. Pai, *Energy Function Analysis for Power System Stability*. Kluwer Academic Publishers, Boston, 1989.

[79] S. Chakrabarti and E. Kyriakides, "Optimal placement of phasor measurement units for power system observability," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1433–1440, Aug. 2008.

[80] Powertech Labs, "DSATools: Dynamic Security Assessment Software," [Online] Available: `http://www.dsatools.com`.

[81] M. He, V. Vittal, and J. Zhang, "Online dynamic security assessment with missing pmu measurements: A data mining approach," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1969–1977, 2013.

[82] Alberta Electric System Operator Rules, "Section 502.9: synchrophasor measurement unit technical requirements," Aug. 2012, [Online] Available: `http://www.aeso.ca/downloads/2012-08-30_Section_502-9_phasor.pdf`.

[83] Schweitzer Engineering Laboratories Technical Report, "Improving the availability of synchrophasor data," Aug. 2011, [Online] Available: `https://www.selinc.com/TheSynchrophasorReport.aspx?id=98004`.

[84] R. Emami and A. Abur, "Robust measurement design by placing synchronized phasor measurements on network branches," *IEEE Trans. on Power Syst.*, vol. 25, no. 1, pp. 38–43, Feb. 2010.

[85] A. Gomez-Exposito, A. Abur, P. Rousseaux, A. de la Villa Jaen, and C. Gomez-Quiles, "On the use of PMUs in power system state estimation," in *Proc. 17th Power Systems Computation Conference*, Stockholm, Sweden, Aug. 2011.

[86] Y. Wang, W. Li, and J. Lu, "Reliability analysis of wide-area measurement system," *IEEE Trans. Power Del.*, vol. 25, no. 3, pp. 1483–1491, July 2010.

[87] R. Bryll, R. Gutierrez-osuna, and F. K. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition*, vol. 36, no. 6, pp. 1291–1302, June 2003.

[88] T. K. Ho, "Random decision forests," in *Proc. Third Intl Conf. Document Analysis and Recognition*, Montreal, Canada, Aug. 1995, pp. 278–282.

[89] A. Phadke and J. Thorp, *Synchronized Phasor Measurements and Their Applications*. New York: Springer, 2008.

[90] T. K. Ho, "Random decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[91] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[92] V. Khiabani, O. P. Yadav, and R. Kavesseri, "Reliability-based placement of phasor measurement units in power systems," *J. Risk and Reliability*, vol. 226, no. 1, pp. 109–117, Feb. 2012.

[93] L. Hyafil and R. L. Rivest, "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters*, vol. 5, no. 1, pp. 15–17, May 1976.

[94] F. Aminifar, S. Bagheri-Shouraki, M. Fotuhi-Firuzabad, and M. Shahidehpour, "Reliability modeling of PMUs using fuzzy sets," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2384–2391, Oct. 2010.

[95] C. Zheng, V. Malbasa, and M. Kezunovic, "Regression tree for stability margin prediction using synchrophasor measurements," 2012, to appear, *IEEE Trans. on Power Syst.*

[96] "IEEE standard for synchrophasors for power systems," 2005, IEEE Std. C37. 118-2005.

[97] R. Leon, V. Vittal, and G. Manimaran, "Application of sensor network for secure electric energy infrastructure," *IEEE Trans. on Power Del.*, vol. 22, no. 2, pp. 1021–1028, April 2007.

[98] A. Phadke, "Synchronized phasor measurements-a historical overview," in *Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES*, vol. 1, Oct. 2002, pp. 476–479.

[99] J. Tate and T. Overbye, "Line outage detection using phasor angle measurements," *IEEE Trans. on Power Syst.*, vol. 23, no. 4, pp. 1644–1652, Nov. 2008.

[100] J. Zhu and A. Abur, "Improvements in network parameter error identification via synchronized phasors," *IEEE Trans. on Power Syst.*, vol. 25, no. 1, pp. 44–50, Feb. 2010.

[101] M. He and J. Zhang, "Fault detection and localization in smart grid: A probabilistic dependence graph approach," in *2010 First IEEE International Conference on SmartGridComm*, Oct. 2010, pp. 43–48.

[102] N. Zhang and M. Kezunovic, "Improving real-time fault analysis and validating relay operations to prevent or mitigate cascading blackouts," in *Transmission and Distribution Conference and Exhibition, 2005/2006 IEEE PES*, May 2006, pp. 847–852.

[103] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, 2008.

[104] M. Luettgen, W. Karl, A. Willsky, and R. Tenney, "Multiscale representations of Markov random fields," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3377–3396, Dec 1993.

[105] A. Anandkumar, L. Tong, and A. Swami, "Detection of GausscMarkov random fields with nearest-neighbor dependency," *IEEE Trans. on Inf. Theory*, vol. 55, no. 2, pp. 816–827, Feb 2009.

[106] S. Krishnamachari and R. Chellappa, "Multiresolution Gauss-Markov random field models for texture segmentation," *IEEE Trans. on Image Processing*, vol. 6, no. 2, pp. 251–267, Feb 1997.

[107] J. M. Hammersley and P. Clifford, "Markov fields on finite graphs and lattices," Unpublished, 1971.

[108] P. Perez and F. Heitz, "Restriction of a Markov random field on a graph and multiresolution statistical image modeling," *IEEE Trans. on Inf. Theory*, vol. 42, no. 1, pp. 180–190, Jan 1996.

[109] P. Perez and F. Heitz, "Restriction of Markov random field on a graph and multiresolution image analysis," INRIA Tech., Rep. 2170, Mar. 1994.

[110] R. Christie, B. Wollenberg, and I. Wangensteen, "Transmission management in the deregulated environment," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 170–195, Feb 2000.

[111] J. Dopazo, O. Klitin, and A. Sasson, "Stochastic load flows," *IEEE Trans. on Power App. Syst.*, vol. 94, no. 2, pp. 299–309, Mar 1975.

[112] A. Schellenberg, W. Rosehart, and J. Aguado, "Cumulant-based probabilistic optimal power flow (p-opf) with Gaussian and Gamma distributions," *IEEE Trans. on Power Syst.*, vol. 20, no. 2, pp. 773–781, May 2005.

[113] A. Meliopoulos, G. Cokkinides, and X. Chao, "A new probabilistic power flow analysis method," *IEEE Trans. on Power Syst.*, vol. 5, no. 1, pp. 182–190, Feb 1990.

[114] M. G. Morgan, J. Apt, L. B. Lave, M. D. Ilic, M. Sirbu, and J. M. Peha, "The many meanings of smart grid," [Online] Available: `http://wpweb2.tepper.cmu.edu/ceic/papers/SmartGrid.htm`.

[115] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall/CRC, 2005.

[116] N. H. Timm, *Applied Multivariate Analysis*. Series: Springer Texts in Statistics, 2002.

[117] D. M. Malioutov, J. K. Johnson, and A. S.Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 1, no. 7, pp. 2031–2064, 2006.

[118] J. Dahl, V. Roychowdhury, and L. Vandenberghe, "Covariance selection for non-chordal graphs via chordal embedding," *Optimization Methods and Software*, vol. 23, no. 4, pp. 501–520, 2005.

[119] A. Dogandzic and B. Zhang, "Distributed estimation and detection for sensor networks using hidden Markov random field models," *IEEE Trans. on Signal Processing*, vol. 54, no. 8, pp. 3200–3215, aug. 2006.

[120] M. Girvan and M. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 1, pp. 7821–7826, 2002.

[121] "Matpower: a Matlab power system simulation package," [Online] Available: `http://www.pserc.cornell.edu/matpower`.

[122] "System protection schemes in power networks," in *International Conference on Large High Voltage Electric Systems (CIGRE)*, Jan 2001.

[123] C. Ferreira, J. Gama, V. Miranda, and A. Botterud, "A survey on wind power ramp forecasting," in *Argonne National Laboratory Report ANL/DIS 10-13*, Dec 2010.

[124] "Guidelines for smart grid cyber security: vol. 2, privacy and the smart grid," [Online] Available: `http://csrc.nist.gov/publications/nistir/ir7628/nistir-7628_vol2.pdf`.

[125] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting algorithms as gradient descent," in *Neural Inf. Process. Syst.*, 1999, pp. 512–518.

APPENDIX A

DERIVATIONS FOR CHAPTER 3

A.1 Proof of Proposition 3.3.1

D_t and W usually have continuous, symmetrical and unimodal probability distributions. Since $\sigma_Y^2 \gg \sigma_o^2$, intuitively, there exists a finite constant $c_0 \geq 0$, such that:

$$\mathbf{P}\left(\{Y - q_o(v) \leq -c_0\sigma_o(v)\} \cup \{Y - q_o(v) \geq c_0\sigma_o(v)\}\right) \approx 1, \tag{A.1}$$

and

$$Q(-c_0) \approx 1, \quad Q(c_0) \approx 0, \quad \exp(-c_0^2/2) \approx 0. \tag{A.2}$$

If $Y - q_o(v) \geq c_0\sigma_o(v)$, (3.13) boils down to:

$$\begin{aligned}\tilde{R}^l(\psi^l, s, u, v) = (v - (c_1 - c_p))\, q_o(v) + (c_1 - c_p)Y \\ + uD_t - c_1 s;\end{aligned} \tag{A.3}$$

When $Y - q_o(v) \leq -c_0\sigma_o(v)$, (3.13) simplifies to

$$\tilde{R}^l(\psi^l, s, u, v) = (v - c_2)\, q_o(v) + uD_t - c_1 s + c_2 Y. \tag{A.4}$$

It is clear that (A.3) and (A.4) are unimodal for $v \in [v_{min}, v_{cap}]$, both with peaks at $v = v_{cap}$. This yields the real-time pricing policy: $\tilde{\vartheta}_{s,u}(\psi^l) = v_{cap}$.

A.2 Proof of Proposition 3.3.2

When the demand of non-persistent opportunistic energy users is relatively elastic, i.e., $\gamma_o < -1$, $\sigma_o(v)$ can be expected to be much smaller than that in the inelastic case under the same real-time prices. With this insight, we resort to the "certainty equivalence" techniques. By approximating D_o by its mean $q_o(v)$, the net profit is given by:

$$\begin{aligned}\hat{R}^l(\psi^l, s, u, v) = uD_t - c_1 s + c_2 Y + (v - c_2)\, q_o(v) \\ -c(Y - q_o(v))^+.\end{aligned} \tag{A.5}$$

Clearly, (A.5) is a piece-wise polynomial in v. Thus, the optimal v could be easily obtained, as summarized in Proposition 3.3.2.

A.3 Proof of Proposition 3.3.3

We first show that $\tilde{\vartheta}_{s,u}$ depend on S and u only through $s - \mathbf{E}^u_{D_t}[D_t]$. For convenience, define $s' = s - \mathbf{E}^u_{D_t}[D_t]$. Since $Y = s' + W - \varepsilon_t$, it is clear from Proposition 3.3.1 and Proposition 3.3.2 that the real-time pricing policy $\tilde{\vartheta}_{s,u}$ depends on the day-ahead decision only through s'. We denote this policy as $\tilde{\vartheta}_{s'} : (W, \varepsilon_t) \to v$. With this insight, by using the change of variable technique in (3.13), the objective function of $\tilde{\mathcal{P}}^{\mathbf{DA}}_{\mathbf{non-pst}}$ can be rewritten as

$$\mathbf{E}_W \mathbf{E}^u_{D_t}\left[R^l(\psi^l, s', u, \tilde{\vartheta}_{s'}(W, \varepsilon_t))\right] = f_1(u) + f_2(s') + c_2 \mathbf{E}_W[W], \tag{A.6}$$

where

$$f_1(u) \stackrel{\Delta}{=} \alpha_t(u - c_1)u^{\gamma_t}, \tag{A.7}$$

$$f_2(s') \stackrel{\Delta}{=} \mathbf{E}_{W,\varepsilon_t}\mathbf{E}_{D_o}^{\tilde{\vartheta}_{s'}(W,\varepsilon_t)}\big[(\tilde{\vartheta}_{s'}(W,\varepsilon_t) - c_2)D_o - c\mathbf{1}_B\left(s' - \varepsilon_t - D_o + W\right)\big] + (c_2 - c_1)s'. \tag{A.8}$$

Let $\mathcal{F}$ denote the solution space for the objective function of $\tilde{\mathcal{P}}^{\mathbf{DA}}_{\mathbf{non-pst}}$ defined in (A.6). Then,

$$\mathcal{F} \;=\; \{(u, s'); u \geq 0, s' \geq -\alpha_t u^{\gamma_t}\}. \tag{A.9}$$

It can be verified that u^* defined in the proposition statement maximizes $f_1(u)$. Define $s'_0 = -\alpha_t u^{*\gamma_t}$, and let s'^* maximize $f_2(s')$. If we show that (u^*, s'^*) belongs to the solution space $\mathcal{F}$, then (u^*, s'^*) optimizes the day-ahead scheduling problem in (A.6). Since $u^* \geq 0$, it is now sufficient to show that $s'^* \geq s'_0$. A sufficient condition to establish this is given by

$$f_2(s') \leq f_2(s'_0),\ \forall\ s' \leq s'_0. \tag{A.10}$$

Under condition **A**, wind generation is not sufficient to meet the total energy demand when day-ahead price is u^*, thus:

$$W < (-s'_0 + \varepsilon_t) + D_o. \tag{A.11}$$

Therefore,

$$W + s' < \varepsilon_t + D_o,\ \forall\ s' \leq s'_0. \tag{A.12}$$

It follows that 1) $W + s < D_t + D_o$, i.e., there is no scheduled energy surplus, thus $\mathbf{1}_B = 0$ in (A.6); 2) Using the preceding statement, recalling the definition of Y, we see that $Y < q_o(\frac{\gamma_o c_2}{1+\gamma_o})$. Thus, from Proposition 3.3.2, the optimal real-time price $\vartheta_{s'}(W, \varepsilon_t)$ turns out to be a constant $\frac{\gamma_o c_2}{1+\gamma_o}$, i.e., independent of the system state and the day-ahead decisions, when the opportunistic energy users are relatively elastic. Also, for the relatively inelastic case, we know from Proposition 3.3.1 that the optimal real-time price is a constant v_{cap}. Letting $v_0(\gamma_o)$ denote this constant real-time price for both the elastic and inelastic cases, respectively, we have

$$f_2(s') = (c_2 - c_1)s' + (v_0(\gamma_o) - c_2)q_o(v_0(\gamma_o)),\ \forall\ s' \leq s'_0. \tag{A.13}$$

Therefore, $f_2(s') \leq f_2(s'_0),\ \forall\ s' \leq s'_0$, and (u^*, s'^*) indeed lies in the feasible region $\mathcal{F}$ and hence optimizes the day-ahead scheduling problem in (A.6). The optimal day-ahead decision, S^*, can now be computed using s'^* and u^*.

APPENDIX B

DERIVATIONS FOR CHAPTER 4

B.1 Derivation for $\hat{\mathrm{e}}(H_L) \leq C_N(F_L)$

By definition, the misclassification error rate of H_L is given by:

$$\hat{\mathrm{e}}(H_L) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}_{\{y_n \neq H_L(\mathbf{x}_n)\}}. \tag{B.1}$$

In the case that $y_n = H_L(\mathbf{x}_n)$,

$$\mathbf{1}_{\{y_n \neq H_L(\mathbf{x}_n)\}} = 0 < \log_2(1 + \mathrm{e}^{-y_n F_L(\mathbf{x}_n)}). \tag{B.2}$$

And in the case that $y_n \neq H_L(\mathbf{x}_n)$, it is easy to see that $y_n F_L(\mathbf{x}_n) < 0$. Then,

$$\mathbf{1}_{\{y_n \neq H_L(\mathbf{x}_n)\}} = 1 < \log_2(1+1) < \log_2(1 + \mathrm{e}^{-y_n F_L(\mathbf{x}_n)}). \tag{B.3}$$

In sum, we have

$$\hat{\mathrm{e}}(H_L) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}_{\{y_n \neq H_L(\mathbf{x}_n)\}} < \frac{1}{N}\sum_{n=1}^{N} \log_2(1 + \mathrm{e}^{-y_n F_L(\mathbf{x}_n)}) = C_N(F_L). \tag{B.4}$$

B.2 Derivation for Convexity of $C_N(F_L)$

Let F_L and F'_L be two functions in the linear closure of $\mathcal{H}_J$. For $\alpha \in [0,1]$,

$$C_N(\alpha F_L + (1-\alpha)F'_L) = \frac{1}{N}\sum_{n=1}^{N} \log_2(1 + \mathrm{e}^{-y_n(\alpha F_L(\mathbf{x}_n) + (1-\alpha)F'_L(\mathbf{x}_n))}) \tag{B.5}$$

It is easy to see that $\log_2(1+\mathrm{e}^{-x})$ is convex with regard to x. Then,

$$\begin{aligned}&\log_2(1 + \mathrm{e}^{-y_n(\alpha F_L(\mathbf{x}_n) + (1-\alpha)F'_L(\mathbf{x}_n))}) \leq \\ &\qquad \alpha \log_2(1 + \mathrm{e}^{-y_n F_L(\mathbf{x}_n)}) + (1-\alpha)\log_2(1 + \mathrm{e}^{-y_n F'_L(\mathbf{x}_n)}).\end{aligned} \tag{B.6}$$

Combining the above two equations, we have

$$\begin{aligned}&C_N(\alpha F_L + (1-\alpha)F'_L) \\ &\leq \frac{1}{N}\sum_{n=1}^{N} \alpha \log_2(1 + \mathrm{e}^{-y_n F_L(\mathbf{x}_n)}) + \frac{1}{N}\sum_{n=1}^{N}(1-\alpha)\log_2(1 + \mathrm{e}^{-y_n F'_L(\mathbf{x}_n)}) \\ &= \alpha C_N(F_L) + (1-\alpha)C_N(F'_L).\end{aligned} \tag{B.7}$$

Therefore, $C_N(F_L)$ is convex.

B.3 Derivation for Differentiability of $C_N(F_L)$

Using tools of functional derivatives, the derivative of $C_N(F_L)$ is given by

$$\begin{aligned}
\nabla C_N(F_L) &\triangleq \frac{\mathrm{d}}{\mathrm{d}\alpha} C_N(F_L + \alpha \mathbf{1}_{\mathbf{x}}) \mid_{\alpha=0} \\
&= \lim_{\alpha \to 0} \frac{1}{\alpha} (C_N(F_L + \alpha \mathbf{1}_{\mathbf{x}}) - C_N(F_L)) \\
&= \lim_{\alpha \to 0} \frac{1}{\alpha} \frac{1}{N} \sum_{n=1}^{N} \log_2 \left(\frac{1 + \mathrm{e}^{-y_n \left(F_L(\mathbf{x}_n) + \mathbf{1}_{\{\mathbf{x}=\mathbf{x}_n\}} \right)}}{1 + \mathrm{e}^{-y_n F_L(\mathbf{x}_n)}} \right) \\
&= \frac{1}{N} \sum_{n=1}^{N} \frac{\mathrm{d}}{\mathrm{d}\alpha} \log_2 \left(1 + \mathrm{e}^{-y_n \left(F_L(\mathbf{x}_n) + \mathbf{1}_{\{\mathbf{x}=\mathbf{x}_n\}} \right)} \right) \mid_{\alpha=0} \\
&= -\frac{1}{N \ln 2} \sum_{n=1}^{N} \frac{y_n \mathbf{1}_{\{\mathbf{x}=\mathbf{x}_n\}}}{1 + \mathrm{e}^{y_n F_L(\mathbf{x}_n)}}. \qquad \text{(B.8)}
\end{aligned}$$

Therefore, $C_N(F_L)$ is differentiable. Further, for any two functions F_L and F'_L in the linear closure of $\mathcal{H}_J$, we have

$$\begin{aligned}
&\parallel \nabla C_N(F_L) - \nabla C_N(F'_L) \parallel \\
=&\parallel -\frac{1}{N \ln 2} \sum_{n=1}^{N} y_n \mathbf{1}_{\{\mathbf{x}=\mathbf{x}_n\}} \left(\frac{1}{1 + \mathrm{e}^{y_n F_L(\mathbf{x}_n)}} - \frac{1}{1 + \mathrm{e}^{y_n F'_L(\mathbf{x}_n)}} \right) \parallel \\
=& \frac{1}{N \ln 2} \max_n \mid \frac{1}{1 + \mathrm{e}^{y_n F_L(\mathbf{x}_n)}} - \frac{1}{1 + \mathrm{e}^{y_n F'_L(\mathbf{x}_n)}} \mid \qquad \text{(B.9)}
\end{aligned}$$

It is clear that $\frac{1}{1+\mathrm{e}^x}$ is Lipschitz continuous with regard to x with Lipschitz constant 1. Thus,

$$\begin{aligned}
&\mid \frac{1}{1 + \mathrm{e}^{y_n F_L(\mathbf{x}_n)}} - \frac{1}{1 + \mathrm{e}^{y_n F'_L(\mathbf{x}_n)}} \mid \leq \mid y_n F_L(\mathbf{x}_n) - y_n F'_L(\mathbf{x}_n) \mid = \\
&\qquad \mid F_L(\mathbf{x}_n) - F'_L(\mathbf{x}_n) \mid \leq \parallel F_L - F'_L \parallel, \qquad \text{(B.10)}
\end{aligned}$$

Therefore,

$$\parallel \nabla C_N(F_L) - \nabla C_N(F'_L) \parallel \leq \frac{1}{N \ln 2} \parallel F_L - F'_L \parallel, \qquad \text{(B.11)}$$

i.e., $C_N(F_L)$ is Lipschitz differentiable with Lipschitz constant $\frac{1}{N \ln 2}$.

B.4 Derivation for Convergence of F_L

Using Lemma 3 of [125], for any $\alpha \leq 0$, we have

$$\begin{aligned} C_N(F_l) - C_N(F_{l+1}) &\geq C_N(F_l) - C_N(F_l + \alpha h_{l+1}) \\ &\geq -\alpha\langle \triangledown C_N(F_l), h_{l+1}\rangle - \frac{1}{2N\ln 2}\alpha^2 \parallel h_{l+1} \parallel^2 \\ &\geq \frac{N\ln 2\langle \triangledown C_N(F_l), h_{l+1}\rangle^2}{2 \parallel h_{l+1} \parallel^2}. \end{aligned} \tag{B.12}$$

Therefore, the boosting process either halts on a finite round L^* with $\langle \triangledown C_N(F_{L^*}), h_{L^*+1}\rangle \geq 0$, or F_L would converge to some function in the linear closure of $\mathcal{H}_J$ with $\lim\limits_{L\to\infty} \langle \triangledown C_N(F_L), h_{L+1}\rangle = 0$.

B.5 Derivation for Uniqueness of the minimum of $G_N^{(l)}(a)$

Based on (4.3), the derivatives of $G_N^{(l)}(a)$ are given by:

$$\frac{\mathrm{d}}{\mathrm{d}a}G_N^{(l)}(a) = -\frac{1}{N\ln 2}\sum_{n=1}^{N}\frac{y_n h_l(\mathbf{x}_n)}{1+\mathrm{e}^{y_n(F_{l-1}(\mathbf{x}_n)+a h_l(\mathbf{x}_n))}}, \tag{B.13}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}a^2}G_N^{(l)}(a) = \frac{1}{N\ln 2}\sum_{n=1}^{N}\frac{\mathrm{e}^{F_{l-1}(\mathbf{x}_n)+a h_l(\mathbf{x}_n)}}{\left(1+\mathrm{e}^{F_{l-1}(\mathbf{x}_n)+a h_l(\mathbf{x}_n)}\right)^2}. \tag{B.14}$$

Under the condition that h_l is a "descent direction" of $C_N(F_{l-1})$, i.e.,

$$\langle h_l, \triangledown C_N(F_{l-1})\rangle = -\frac{1}{N\ln 2}\sum_{n=1}^{N}\frac{y_n h_l(\mathbf{x}_n)}{1+\mathrm{e}^{y_n F_{l-1}(\mathbf{x}_n)}} < 0. \tag{B.15}$$

It thus follows that $G_N^{(l)'}(0) < 0$. Further, it is clear from (B.14) that $G_N^{(l)''}(a) > 0$ holds for any $a \in \mathcal{R}^+$. Therefore, $G_N^{(l)}(a)$ has a unique minimum a^* in $\mathcal{R}^+$, at which $G_N^{(l)'}(a^*) = 0$.

APPENDIX C

DERIVATIONS FOR CHAPTER 6

C.1 Proof of Lemma 6.3.1

In what follows, by leveraging the Hammersley-Clifford Theorem [107], we first explore the dependency graph and potential functions of the additional sub-field $\mathbf{X}_B$. Following the approach in [108], we define

$$\begin{aligned} \mathcal{C}_B &= \{c \in \mathcal{C} : c \subset B\}, \\ \mathcal{C}_k &= \{c \in \mathcal{C} : c \cap I_k \neq \Phi\}, \end{aligned} \tag{C.1}$$

for $k = 1, 2, \cdots, K$. That is, any clique in $\mathcal{C}_B$ is comprised of border sites only, and any clique in $\mathcal{C}_k$ would contain an inner site in I_k. Further, $\mathcal{C} = \mathcal{C}_B \cup \left(\overset{K}{\underset{k=1}{\cup}} \mathcal{C}_k \right)$. Then, for a configuration $\mathbf{x_B}$ on $\mathbf{B}$, its probability is given by

$$P_{\mathbf{X}_B}(\mathbf{x}_B) = \int_{\lambda \in \Lambda^{S \backslash B}} \frac{1}{Z} \exp\left\{ - \sum_{c \in \mathcal{C}} Vc(\mathbf{x}_B, \lambda) \right\} d\lambda. \tag{C.2}$$

Note that for any $c \in \mathcal{C}_B$, the potential function does not depend on configuration on $S \backslash B$, and for any $c \in \mathcal{C}_k$, its potential function depends only on $\mathbf{X}_{I_k}$. It follows that, (C.2) can be rewritten as

$$\begin{aligned} P_{\mathbf{X}_B}(\mathbf{x}_B) = \frac{1}{Z} \exp\left\{ - \sum_{c \in \mathcal{C}_B} Vc(\mathbf{x}_B, \mathbf{a}_{\bar{B}}) \right\} \\ \prod_{k=1}^{K} \int_{\lambda \in \Lambda^{I_k}} \exp\left\{ - \sum_{c \in \mathcal{C}_k} Vc\left(\mathbf{x}_B, \lambda, \mathbf{a}_{\bar{B} \backslash I_k}\right) \right\} d\lambda, \end{aligned} \tag{C.3}$$

where $\mathbf{a}_{\bar{B}} \in \Lambda^{S \backslash B}$ and $\mathbf{a}_{\bar{B} \backslash I_k} \in \Lambda^{S \backslash \{B \cup I_k\}}$ are arbitrary configurations on $\mathbf{X}_{S \backslash B}$ and $\mathbf{X}_{S \backslash \{B \cup I_k\}}$, respectively. For $k = 1, 2, \cdots, K$, define:

$$V_k(\mathbf{x}_B) = -\ln \int_{\lambda \in \Lambda^{I_k}} \exp\left\{ - \sum_{c \in \mathcal{C}_k} Vc\left(\mathbf{x}_B, \lambda, \mathbf{a}_{\bar{B} \backslash I_k}\right) \right\} d\lambda. \tag{C.4}$$

Then, (C.3) can be rewritten as

$$P_{\mathbf{X}_B}(\mathbf{x}_B) = \frac{1}{Z} \exp\left\{ - \sum_{c \in \mathcal{C}_B} Vc(\mathbf{x}_B, \mathbf{a}_{\bar{B}}) - \sum_{k=1}^{K} V_k(\mathbf{x}_B) \right\}. \tag{C.5}$$

Note that based on the definition in (C.4), potential function $V_k(\mathbf{x}_B)$ depends only on the configuration on the border sites in B_k.

" $\Rightarrow$ ". We prove the contrapositive, i.e., *if $i \not\sim j$ in G, then $i \not\sim j$ in $\tilde{G}_B$.* For notational simplicity, we use $\mathbf{X}_{-i}$ to denote $\mathbf{X}_{B \backslash i}$, and $\mathbf{x}_{-i}$ the corresponding

configuration. By definition, $i \not\sim j$ in $\tilde{G}_B$ refers to that for any $x_i, x_j \in \Lambda$ and $\mathbf{x}_{-ij} \in \Lambda^{B\setminus\{i,j\}}$:

$$\Pr\{X_i = x_i, X_j = x_j | \mathbf{X}_{-ij} = \mathbf{x}_{-ij}\} =$$

$$\Pr\{X_i = x_i | \mathbf{X}_{-ij} = \mathbf{x}_{-ij}\} \Pr\{X_j = x_j | \mathbf{X}_{-ij} = \mathbf{x}_{-ij}\}. \tag{C.6}$$

Under the positiveness condition, it suffices to show that

$$\Pr\{X_i = x_i, X_j = x_j | \mathbf{X}_{-ij} = \mathbf{x}_{-ij}\} =$$

$$\Pr\{X_i = x_i | \mathbf{X}_{-i} = \mathbf{x}_{-i}\} \Pr\{X_j = x_j | \mathbf{X}_{-j} = \mathbf{x}_{-j}\}. \tag{C.7}$$

Without loss of generality, we assume that $i \in B_m$ and $j \in B_q$ with $m, q \in \{1, 2, \cdots, K\}$ and $m \neq q$. Based on (C.5), the right hand side and the left hand side of (C.7) are expanded further as in (C.8).

$$\Pr\{X_i = x_i, X_j = x_j | \mathbf{X}_{-ij} = \mathbf{x}_{-ij}\} =$$

$$\frac{1}{Z_{ij}(\mathbf{x}_{-ij})} \exp\left\{ - \sum_{c \in \mathcal{C}_B : i \in c \text{ or } j \in c} Vc(\mathbf{x}_B, \mathbf{a}_{\bar{B}}) - V_m(\mathbf{x}_B) - V_q(\mathbf{x}_B) \right\},$$

$$Z_{ij}(\mathbf{x}_{-ij}) =$$

$$\int_{\lambda \in \Lambda^2} \exp\left\{ - \sum_{c \in \mathcal{C}_B : i \in c \text{ or } j \in c} Vc(\mathbf{x}_{-ij}, \lambda, \mathbf{a}_{\bar{B}}) - V_m(\mathbf{x}_{-ij}, \lambda) - V_q(\mathbf{x}_{-ij}, \lambda) \right\} d\lambda,$$

$$\Pr\{X_i = x_i | \mathbf{X}_{-i} = \mathbf{x}_{-i}\} = \frac{1}{Z_i(\mathbf{x}_{-i})} \exp\left\{ - \sum_{c \in \mathcal{C}_B : i \in c} Vc(\mathbf{x}_B, \mathbf{a}_{\bar{B}}) - V_m(\mathbf{x}_B) \right\},$$

$$Z_i(\mathbf{x}_{-i}) = \int_{\lambda \in \Lambda} \exp\left\{ - \sum_{c \in \mathcal{C}_B : i \in c} Vc(\mathbf{x}_{-i}, \lambda, \mathbf{a}_{\bar{B}}) - V_m(\mathbf{x}_{-i}, \lambda) \right\} d\lambda,$$

$$\Pr\{X_j = x_j | \mathbf{X}_{-j} = \mathbf{x}_{-j}\} = \frac{1}{Z_j(\mathbf{x}_{-j})} \exp\left\{ - \sum_{c \in \mathcal{C}_B : j \in c} Vc(\mathbf{x}_B, \mathbf{a}_{\bar{B}}) - V_q(\mathbf{x}_B) \right\},$$

$$Z_j(\mathbf{x}_{-j}) = \int_{\lambda \in \Lambda} \exp\left\{ - \sum_{c \in \mathcal{C}_B : j \in c} Vc(\mathbf{x}_{-j}, \lambda, \mathbf{a}_{\bar{B}}) - V_q(\mathbf{x}_{-j}, \lambda) \right\} d\lambda. \tag{C.8}$$

Plugging (C.8) in (C.7), it follows that a necessary and sufficient condition for (C.7) to hold is:

$$\frac{Z_{ij}(x_{-ij})}{Z_i(x_{-i}) Z_j(x_{-j})} \exp\left\{ - \sum_{\{i,j\} \subseteq c} Vc(x_B, \mathbf{a}_{\bar{B}}) \right\} = 1. \tag{C.9}$$

Note that if $i \not\sim j$ in G, then there are no cliques in G that contain both i and j.

Actually, the set $\{c \in \mathcal{C}_B : \{i,j\} \subseteq c\}$ is the null set. In this scenario, it is easy to show that $Z_{ij}(x_{-ij}) = Z_i(x_{-i}) Z_j(x_{-j})$, since $\{c \in \mathcal{C}_B : i \in c\}$ and $\{c \in \mathcal{C}_B : j \in c\}$ are disjoint. Thus, (C.9), (C.7), and (C.6) hold, i.e., the pairwise Markov property between site i and j in the additional sub-field $\mathbf{X}_B$ is proved .

" $\Leftarrow$ ". We prove this part by contradiction. If $i \sim j$ in G, according to Proposition 3 of [108], there exists a clique $c_0 \in \mathcal{C}$ such that $\{i,j\} \subseteq c_0$ and Vc_0 is non-zero and irreducible. Further, since sites i and j are contained in different border sets, all the sites in c_0 are the border sites of $\mathbf{X}$. Therefore, we have $c_0 \subset \mathbf{X_B}$.

On the other hand, since $i \not\sim j$ in $\tilde{G}_B$, then c is not a clique of $\tilde{G}_B$. According to the property of GMRF $\mathbf{X}$ as a Gibbs random field, potential functions on non-clique c_0 is zero everywhere on Λ^B. However, observe from (C.5) that the potential function $V_k(\mathbf{x}_B)$ only depends on the sites in B_k, thus, the potential function on non-clique c in $\tilde{G}_B$ is exactly equal to that on clique c in G with any given configuration $\mathbf{a}_{\bar{B}}$ on $S \backslash B$. It follows that $Vc_0(\mathbf{x}_B, \mathbf{a}_{\bar{B}})$ is zero for any $\mathbf{x}_B \in \Lambda^B$, which contradicts to the fact that Vc_0 is non-zero and irreducible. Therefore, the contradiction is proved, i.e., if $i \sim j$ in G, then $i \sim j$ in $\tilde{G}_B$.

If a clique c of G contains two sites that are connected by a tie-line edge, then all the sites in c are border sites. Further, based on Lemma 6.3.1(a), we conclude that c is also a clique of $\tilde{G}_B$.

Now, based on (C.5), we express $P_{\mathbf{X}_B}(\mathbf{x}_B)$ using only non-zero irreducible potential functions. Further, since Vc is non-zero and irreducible, therefore, the probability distribution of $\mathbf{X}_B$ could be expressed in terms of the potential function on c, denoted as $\tilde{V}c$. Note that in (C.5), the potential function V_k only depends on the sites in B_k. Thus, it is easy to show that $\tilde{V}c$ is irreducible and, $\tilde{V}c(\mathbf{x}_B) = Vc(\mathbf{x}_B, \mathbf{a}_{\bar{B}})$ for any given configuration $\mathbf{x}_B$ on $\mathbf{X}_B$,

C.2 Proof of Lemma 6.3.2

Similarly, we first explore the dependency graph and the clique potential functions of the sub-field $\mathbf{X}_{S_k}$, $k \in \{1, 2, \cdots, K\}$. Define $\mathcal{C}_{\bar{I}_k} = \mathcal{C} \backslash \mathcal{C}_{I_k}$. Further, the probability of a configuration $\mathbf{x}_{S_k}$ on $\mathbf{X}_{S_k}$ is given by

$$P_{\mathbf{X}_{S_k}}(\mathbf{x}_{S_k}) = \int_{\lambda \in \Lambda^{S \backslash S_k}} \frac{1}{Z} \exp\left\{ - \sum_{c \in \mathcal{C}} Vc(\mathbf{x}_{S_k}, \lambda) \right\} d\lambda. \tag{C.10}$$

Observe that if $c \in \mathcal{C}_{I_k}$, then all the sits in c is contained in S_k, this is because inner sites are not connected to other sub-field. Therefore, the potential function of c does not depend on the configuration on $S \backslash S_k$. Then, (C.10) could be rewritten as

$$P_{\mathbf{X}_{S_k}}(\mathbf{x}_{S_k}) = \frac{1}{Z} \exp\left\{ - \sum_{c \in \mathcal{C}_{I_k}} Vc\left(\mathbf{x}_{S_k}, \mathbf{a}_{\bar{S}_k}\right) \right\}$$

$$\int_{\lambda\in\Lambda^{S\setminus S_k}} \exp\left\{-\sum_{c\in\mathcal{C}_{\bar{I}_k}} Vc\left(\mathbf{x}_{S_k},\lambda\right)\right\}d\lambda, \tag{C.11}$$

where, $\mathbf{a}_{\bar{S}_k}\in\Lambda^{S\setminus S_k}$ is any given configuration on $\mathbf{X}_{S\setminus S_k}$. For convenience, define:

$$\hat{V}_k\left(\mathbf{x}_{S_k}\right) = -\ln\int_{\lambda\in\Lambda^{S\setminus S_k}} \exp\left\{-\sum_{c\in\mathcal{C}_{\bar{I}_k}} Vc\left(\mathbf{x}_{S_k},\lambda\right)\right\}d\lambda. \tag{C.12}$$

It follows that in this case, (C.2) can be rewritten as

$$P_{\mathbf{X}_{S_k}}\left(\mathbf{x}_{S_k}\right) = \frac{1}{Z}\exp\left\{-\sum_{c\in\mathcal{C}_{I_k}} Vc\left(\mathbf{x}_{S_k},\mathbf{a}_{\bar{S}_k}\right) - \hat{V}_k\left(\mathbf{x}_{S_k}\right)\right\}. \tag{C.13}$$

Observe from (C.12) that $\hat{V}_k(\mathbf{x}_{S_k})$ depends only on the configuration on $\mathbf{X}_{B_k}$. Using (C.13), we can complete the proof of the lemma using a similar procedure as in that of Lemma 6.3.1.

C.3 Proof of Proposition 6.3.1

Recall that $\mathbf{X}_B$ denotes the additional sub-field, in which the border sites are arranged in accordance with the sub-fields. Then, according to [109], the information matrix $\mathbf{J}_B$ is given by

$$\mathbf{J}_B = \mathbf{J}_{BB} - \mathbf{J}_{BI}\mathbf{J}_{II}^{-1}\mathbf{J}_{BI}^T. \tag{C.14}$$

It is clear that the off-diagonal blocks of $\mathbf{J}_{BI}$ are zero matrices, because a border site and an inner site are not connected if they are contained in different sub-fields. Similarly, $\mathbf{J}_{II}$ is also block diagonal. Therefore,

$$\begin{aligned}\mathbf{J}_{BI} &= \mathbf{diag}\left([\mathbf{J}_{BI}]_{11},\cdots,[\mathbf{J}_{BI}]_{KK}\right),\\ \mathbf{J}_{II} &= \mathbf{diag}\left([\mathbf{J}_{II}]_{11},\cdots,[\mathbf{J}_{II}]_{KK}\right).\end{aligned} \tag{C.15}$$

Plugging (C.15) in (C.14), we obtain that

$$[\mathbf{J}_B]_{kk} = [\mathbf{J}_{BB}]_{kk} - [\mathbf{J}_{BI}]_{kk}[\mathbf{J}_{II}]_{kk}^{-1}[\mathbf{J}_{BI}]_{kk}^T, \tag{C.16}$$

for $k=1,\cdots,K$. Then Proposition 6.3.1 follows.