2016

Toward Intelligent Cyber-Physical Systems: Algorithms, Architectures, and Applications

Bo Tang

University of Rhode Island, botang1985@gmail.com

Follow this and additional works at: https://digitalcommons.uri.edu/oa_diss

Recommended Citation

Tang, Bo, "Toward Intelligent Cyber-Physical Systems: Algorithms, Architectures, and Applications" (2016). *Open Access Dissertations*. Paper 481.
https://digitalcommons.uri.edu/oa_diss/481

This Dissertation is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.

TOWARD INTELLIGENT CYBER-PHYSICAL SYSTEMS: ALGORITHMS,
ARCHITECTURES, AND APPLICATIONS

BY
BO TANG

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2016

DOCTOR OF PHILOSOPHY DISSERTATION
OF
BO TANG

APPROVED:

Dissertation Committee:
Major Professor

Haibo He

Steven Kay

Lisa DiPippo

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2016

ABSTRACT

Cyber-physical systems (CPS) are the new generation of engineered systems integrated with computation and physical processes. The integration of computation, communication and control adds new capabilities to the systems being able to interact with physical world. The uncertainty in physical environment makes future CPS to be more reliant on machine learning algorithms which can learn and accumulate knowledge from historical data to support intelligent decision making. Such CPS with the incorporation of intelligence or smartness are termed as intelligent CPS which are safer, more reliable and more efficient.

This thesis studies fundamental machine learning algorithms in supervised and unsupervised manners and examines new computing architecture for the development of next generation CPS. Two important applications of CPS, including smart pipeline and smart grid, are also studied in this thesis. Particularly, regarding supervised machine learning algorithms, several generative learning and discriminative learning methods are proposed to improve learning performance. For the generative learning, we build novel classification methods based on exponentially embedded families (EEF), a new probability density function (PDF) estimation method, when some of the sufficient statistics are known. For the discriminative learning, we develop an extended nearest neighbor (ENN) method to predict patterns according to the maximum gain of intra-class coherence. The new method makes a prediction in a “two-way communication” style: it considers not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors. By exploiting the generalized class-wise statistics from all training data, the proposed ENN is able to learn from the global distribution, therefore improving pattern recognition performance and providing a powerful technique for a wide range of data analysis applications. Based on the

concept of ENN, an anomaly detection method is also developed in an unsupervised manner.

CPS usually have high-dimensional data, such as text, video, and other multi-modal sensor data. It is necessary to reduce feature dimensions to facilitate the learning. We propose an optimal feature selection framework which aims to select feature subsets with maximum discrimination capacity. To further address the information loss issue in feature reduction, we develop a novel learning method, termed generalized PDF projection theorem (GPPT), to reconstruct the distribution in high-dimensional raw data space from the low-dimensional feature subspace.

To support the distributed computations throughout the CPS, it needs a novel computing architecture to offer high-performance computing over multiple spatial and temporal scales and to support Internet of Things for machine-to-machine communications. We develop a hierarchical distributed Fog computing architecture for the next generation CPS. A prototype of such architecture for smart pipeline monitoring is implemented to verify its feasibility in real world applications.

Regarding the applications, we examine false data injection detection in smart grid. False data injection is a type of malicious attack which can threaten the security of energy systems. We examine the observability of false data injection and develop statistical models to estimate underlying system states and detect false data injection attacks under different scenarios to enhance the security of power systems.

ACKNOWLEDGMENTS

In the first place, I would like to express my sincere gratitude to my advisor, Prof. Haibo He, for his continuous support, encouragement and inspiration throughout my years at URI. Without these, this thesis could be never possible. He has always given me valuable feedback with the incredible insight of new research directions that was worth pursuing. I really appreciate his encouragement and support on my research, and his willingness to share his deep knowledge with me. It is my fortune to have had him as my PhD advisor.

I am also grateful for my thesis committee members: Prof. Steven Kay, Prof. Tao Wei, Prof. Lisa DiPippo, and Prof. P. V. August. Their precious comments and suggestions have made this thesis to be in better shape. I have been always inspired by Prof. Kay from his books and other publications, and his insightful suggestions and comments for my research. I have truly enjoyed his three courses and discussions with him in every weekly meeting. Also thanks for Prof. Wei and Prof. Qing Yang for his constructive suggestions and collaborations for the Smart City research project.

I would like to thank all group members in the Computational Intelligence and Self-Adaptive System (CISA) Laboratory during my four-years PhD studies at URI: Siyao Fu, Jun Yan, Zhen Ni, Xiangnan Zhong, Jing Wang, Yufei Tang, Zhiqiang Wang, He Jiang, Xueming Hu, Weijun Li, and many other visiting scholars. Thanks to all of you for sharing of your research and knowledge during numerous group and individual discussions. I also would like to thank my close friends within and outside the ECBE department: Quan Ding, Xiang Liu, Yihai Zhu, Brian Chen, Ke Bian, among others, to make my URI's life so colorful and enjoyable.

Finally, I would like to thank my parents and brother for everything they have

done for me. They have always given me freedom and encouragement to build my own career path. I am also blessed to have companionship of my beloved wife, Weiwei Xie, with me throughout these years at URI. Thank you for always being my side, inspiring me, encouraging me and moving our life forward. Thank you for everything.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iv
TABLE OF CONTENTS	vi
LIST OF TABLES	xi
LIST OF FIGURES	xii
CHAPTER	
1 Introduction	1
1.1 Intelligent Cyber-Physical Systems	3
1.1.1 Typical Structure of Cyber-Physical Systems	3
1.1.2 Need for CPS Intelligence	4
1.1.3 Computing and Communication Architecture in CPS	4
1.2 Machine Learning Algorithms in CPS	5
1.3 High-Performance Computing Architectures in CPS	7
1.4 Smart-X applications	8
1.5 Summary of Thesis Contributions	9
1.6 Related Work	13
1.7 Thesis Organization	17
2 Feature Representation	19
2.1 Introduction	19
2.2 Theoretical Framework of Optimal Feature Selection	21

	Page
2.2.1 Divergence Measures for Binary Hypothesis Testing . . .	21
2.2.2 Jeffreys-Multi-Hypothesis Divergence	24
2.2.3 Optimal Feature Subset Selection	25
2.3 Efficient Optimal Feature Subsets Selection for Text Categorization	26
2.3.1 Background of Text Categorization	26
2.3.2 A Greedy Feature Selection Approach	29
2.3.3 An Efficient Feature Selection Approach	33
2.3.4 Multi-class Classification	35
2.3.5 Feature Selection Based on χ^2 Statistics	36
2.4 Experimental Results and Analysis	38
2.4.1 Real-Life Data Sets	38
2.4.2 Results	39
2.5 Summary	40
3 EEF-Based Generative Learning	42
3.1 Introduction to EEF	42
3.2 Properties of EEF	45
3.3 EEF for Classification	45
3.4 Data-Driven Classifier via EEF	47
3.4.1 Hypothesis	47
3.4.2 Distribution Construction via EEF	48
3.4.3 Experiments	50
3.5 Model-Driven Classifier via EEF: Power Quality Disturbance Classification	52
3.5.1 Hypothesis	52

	Page
3.5.2 Reduced Distribution Construction	55
3.5.3 Penalty Version of Classification Rule	56
3.5.4 Simulations	57
3.6 GPPT: Generalized PDF Projection Theorem for Classification based on EEF	58
3.6.1 Motivations	58
3.6.2 GPPT: PDF Construction via EEF	60
3.6.3 GPPT for Classification with Class-Specific Features . .	62
3.6.4 Experimental Results and Analysis	65
3.7 Summary	67
4 ENN-Based Discriminative Learning	68
4.1 ENN: Extended Nearest Neighbor Method for Classification . .	68
4.1.1 Background and Motivations	68
4.1.2 Problem Formulation: Limitations of KNN	69
4.1.3 ENN Classifier	71
4.1.4 ENN.V1 Classifier: An Equivalent Version of ENN	74
4.1.5 ENN.V2 Classifier: An Approximate Version of ENN . .	78
4.1.6 Experiments and Results Analysis	80
4.1.7 Computational Complexity Analysis	85
4.1.8 Discussion	86
4.2 A Local Density-Based Approach with Extended Nearest Neigh- bors for Anomaly Detection	87
4.2.1 Introduction to Anomaly Detection	87
4.2.2 Local Kernel Density Estimation	89

	Page
4.2.3 RDOS: Relative Density-based Outlier Factor	91
4.2.4 Theoretical Properties of RDOS	93
4.2.5 Experimental Results and Analysis	96
4.2.6 Discussion	102
4.3 Summary	102
5 High-Performance Distributed Computing Architectures . . .	104
5.1 Introduction	104
5.2 Hierarchical Distributed Fog Computing Architecture	106
5.3 A Prototype of Fog Computing Architecture for Smart Pipeline	110
5.3.1 Layer 4: Fiber Optic Sensing Networks	110
5.3.2 Layer 3: Edge Device for Feature Extraction	111
5.3.3 Layer 2: Intermediate Computing Node for Event Recog- nition with Spatial-Temporal Association	112
5.3.4 Layer 1: Cloud for Data Management	114
5.3.5 Experimental Results and Analysis	114
5.3.6 Discussion	119
5.4 Summary	121
6 Application: False Data Injection Detection in Smart Grid .	122
6.1 Introduction	122
6.2 Background and Related Work	123
6.3 Proposed State Estimator and False Data Detector with Colored Gaussian Noise	128
6.4 Simulations and Analysis	133
6.5 Summary	135

	Page
7 Conclusions and Future Research Directions	136
7.1 Conclusions	136
7.2 Original Contributions	136
7.3 Future Research Directions	138
LIST OF REFERENCES	141
BIBLIOGRAPHY	155

LIST OF TABLES

Table		Page
3.1	Description of seven UCI data sets	51
3.2	Average testing accuracy and standard derivation of the proposed method, Pseudo MAP, Nearest Neighbor, and Neural Network in percentage. The best value is highlighted with Bold value and the significant improvement is <u>underlined</u> for our proposed method.	52
3.3	Categories of power quality disturbance.	54
3.4	Performance of power quality disturbance using EEF rule compared with the SOLAR, Inductive Inference and C-SVM.	58
4.1	Average testing error rate and standard derivation of ENN ($k = 3$) compared with KNN ($k = 3$), naive Bayes, LDA, and neural network. All results are shown in percentage. For each dataset, we highlight the best result with Bold value among all these five methods. To specifically compare the results of ENN and KNN, we also <u>underline</u> the value if ENN performs significantly better than KNN under one-tailed t -test ($p = 0.01$).	84
4.2	Top $n = 5$ data samples with the largest RDOS in two synthetic data sets.	98
4.3	The characteristics of four data sets	100
4.4	Comparison of detection performance with the metric of AUC of ROC curves for four real-life data sets. For each data set, we highlight the best result with Bold value and the second one with <i>Italic</i> value among all methods.	101
4.5	Comparison of detection performance with the metric of AUC of PR curves for four real-life data sets. For each data set, we highlight the best result with Bold value and the second one with <i>Italic</i> value among all methods.	102

LIST OF FIGURES

Figure		Page
1.1	Typical structure of CPS.	3
1.2	Thesis organization.	18
2.1	Accuracy on the data sets of (a). REUTERS-20 and (b). TDT2, when naive Bayes is used as the classifier.	40
3.1	Flow chart of the GPPT classifier with class-specific features for text categorization (right), compared with the conventional approach (left).	63
3.2	Classification results on REUTERS-20.	66
4.1	One example of the KNN rule in comparison with the Bayesian rule for a two-class classification problem.	71
4.2	(A), (B) and (C) demonstrate the detailed procedure of two versions of ENN classifier. In (A), training samples are stored and their k nearest neighbors and distances are calculated to compute the generalized class-wise statistic T_1 and T_2 . (B) and (C) illustrate the changes of the generalized class-wise statistic for each class, when we iteratively assume Z to be class 1 and class 2. Both of them provide the same classification result, but the recalculation of generalized class-wise statistic T_i^j is avoided in ENN.V1.	77
4.3	Overall classification error rate (in percentage) of ENN and KNN for four Gaussian data models.	81
4.4	Overall classification error rate (in percentage) of ENN and KNN for handwritten digits classification.	82
4.5	Overall classification error rate (in percentage) of ENN and KNN for spam email classification.	83
4.6	Three types of nearest neighbors considered. Arrows from X_i and X_j to $NN_r(X_i)$ and $NN_s(X_j)$, respectively.	91

Figure		Page
4.7	The distribution of normal data and outliers, where the objects: A , B , and C are outliers.	97
4.8	The outlierness scores of all data samples, where the value of RDOS is illustrated by the color and the radius of the circle. . .	97
4.9	The distribution of normal data and outliers, where A , B , C and D are considered as outliers.	98
4.10	The outlierness scores of all data samples, where the value of RDOS is illustrated by the color and the radius of the circle. . .	99
4.11	The normal data and outliers in four real-life data sets: (A) BREAST CANCER, (B) PEN-LOCAL, (C) PEN-GLOBAL, and (D) SATELLITE. Only the first two principle components are shown.	99
5.1	The 4-layer Fog computing architecture in cyber-physical systems.	107
5.2	Data and control flows in hierarchical Fog computing architecture.	109
5.3	Schematic of the sensor interrogation system (TLS: tunable laser source, CPL: coupler, PC: polarization controller, CIR: circulator, BPD: balanced photodiode, DAQ: data acquisition card) . .	111
5.4	Layout of prototype pipeline system.	115
5.5	One example of spatial-temporal event pattern. A colormap is used to represent the temperature.	116
5.6	Fiber optic sensors governed by 6 edge devices. Each edge device is connected to a neighborhood of sensors.	117
5.7	Online inference performance with different number of hidden states in each HMM under different parameter settings: (a). $Q = 2$, $Q = 3$, and $Q = 4$, when two components GMMs are used ($K = 2$); (b). $K = 1$, $K = 2$, and $K = 3$, when two hidden states are used ($Q = 2$).	118
5.8	Online inference performance for different sensor connections to the edge devices.	118

Figure		Page
5.9	Fog vs Cloud computing: (a). The amount of data that are sent to the Cloud per second; (b). The response time for hazardous events, when the Internet bandwidth is $1Mb/s$; (c) The response time for hazardous events with different Internet bandwidths. The log values in the y -axis are used to clearly illustrate the comparisons.	120
6.1	Comparison of ROC curves in observable attacks when: (a) $\sigma^2 = 0.3$, $A = 1$, and $D = 29$; (b) $\sigma^2 = 0.5$, $A = 1$, and $D = 29$; (c) $\sigma^2 = 0.7$, $A = 1$, and $D = 29$	134
6.2	Comparison of ROC curves in ICA-based unobservable attacks when: (a) $\sigma^2 = 0.3$, $\sigma_y^2 = 0.3$, $A = 1$; (b) $\sigma^2 = 0.3$, $\sigma_y^2 = 0.5$, $A = 1$; (c) $\sigma^2 = 0.7$, $\sigma_y^2 = 0.5$, $A = 1$	134

CHAPTER 1

Introduction

In the past few decades, the communications and embedded computing technologies have been widely used in almost all types of systems due to the rapid development of information technologies. The embedding systems which integrate networked computations into physical processes are now referred to as the Cyber-Physical Systems (CPS), which allows researchers to incorporate intelligence or smartness into the systems to enhance the interactions with physical world and enable new functions not previously possible. The CPS are now everywhere: in many consumer products such as robots, vehicles and personal devices; energy systems such as smart grid; many civil infrastructures such as smart building and smart home; industrial control systems such as nuclear plants, to name a few. It has also been seen as the future of information technology which will transform how people interact with the physical world, just as the internet transformed how people interacted with each other.

Toward intelligent CPS, it is necessary to incorporate computational intelligence into physical processes, adding new capacities to the systems such as safety, efficiency and productivity. For example, smart building, or green energy building, can greatly reduce economical cost and environmental impact by maintaining optimal operations under different situations; smart energy systems can be more robust and safer by detecting different types of false data injection attacks; autonomous vehicles can drive safely by avoiding all possible collisions; smart disaster management systems can offer quick response and optimal distribution of municipal emergency services by predicting hazardous events and maintaining situation awareness. In other words, the systems incorporated with the “intelligence” will be

able to handle the uncertainty in physical world such that the systems are robust, adaptive and fault tolerant to the dynamically changing environment.

In this thesis, we contribute to the foundations of computational intelligence in CPS. Specifically, this thesis focuses on the development of advanced machine learning algorithms which could have wide applications in CPS such as disease prediction in smart healthcare systems, attack detection in smart grid and smart infrastructure, event and behavior recognition in smart transportation systems, and so on. Two fundamental machine learning frameworks are proposed: exponentially embedded family (EEF) based learning and extended nearest neighbor (ENN) based learning. Meanwhile, to facilitate the learning in high dimensional data sets, we contribute an effective feature selection method to reduce feature dimensions. In these machine learning algorithms, we aim to extract useful information such as patterns and distributions from the sensor data, which are usually related to the data analysis. However, since the wide deployment of sensors leads to large amounts of data, the traditional computing architecture may be inappropriate for the distributed big data analysis in CPS. We are also interested in investigating a new computing architecture, called Fog computing, for CPS. Two important CPS applications, including smart pipeline and smart grid, are investigated with the incorporation of intelligence. For the rest of this chapter, we will provide a brief introduction to the learning and computing problems in CPS, outline our contributions and provide a summary of related work. We first introduce the general structure of intelligent CPS and discuss the challenges toward intelligent CPS.

1.1 Intelligent Cyber-Physical Systems

1.1.1 Typical Structure of Cyber-Physical Systems

Cyber-physical systems are advanced complex systems in which the sensing, computational and physical processes are tightly coordinated to work together effectively to interact with the physical environment. A general structure of the typical CPS is shown in Fig. 1.1. A typical CPS usually includes four parts: physical components, sensing network, computing devices, and communication network. The *physical components* are the physical systems we are interested to monitor and protect. The second part is the *sensing network* which are consisted of networked sensors. The sensors are necessary to be deployed to monitor the physical environment, e.g., temperature and pressure, and the physical processes, e.g., system states and parameters. The sensor data are transferred to the *computing devices*, in which the data are processed and analyzed and the controllable decisions are made using computational intelligence methods to actuate the physical components with actuators. The computing devices are interconnected by the *communication network* so that they can be coordinated to work effectively to complete desired computational tasks in which spatial data are needed.

Figure 1.1: Typical structure of CPS.

1.1.2 Need for CPS Intelligence

The incorporation of intelligence will make the CPS to be able to execute complex tasks in dynamic environments and under unforeseen conditions. To address this uncertainty of physical world, machine learning provides solutions in a statistic manner, which would always try to make optimal decisions in terms of performance such as prediction accuracy, efficiency, robustness, and other specific metrics. Among many types of machine learning methods, classification and clustering would have wide applications in CPS to identify patterns among the collected sensor data, such as anomaly detection for protecting system safety, behavior recognition for understanding surrounding environment, prediction for system optimization and planning, to name a few. In this thesis, we will propose a wide range of learning algorithms ranging from feature representation, classification and anomaly detection, and show that they can improve the learning performance for many real-life applications.

1.1.3 Computing and Communication Architecture in CPS

The communication among different computing devices are commonly referred to as the Internet of Things (IoT), which is a type of machine-to-machine (M2M) communication. Such M2M communications will be dominated in future CPS, which allows data transfer over a network without or with little human intervention. This transformation of communication network in CPS makes it greatly different from many existing systems with human-to-human or human-to-machine communication.

It is also worth noticing that the wide deployment of sensors in CPS will generate large amounts of data streams, which leads to the *big data analysis* challenge. The big data analysis of CPS requires high-performance distributed computing to process massive data streams in real time. Both IoT and big data analysis require

us to investigate novel computing and communication architectures in CPS. In this thesis, we will propose a new high-performance computing architecture to offer high-performance distributed computing and support the domination of M2M communication in CPS.

1.2 Machine Learning Algorithms in CPS

The first important part of intelligent CPS considered in this thesis is the development of advanced machine learning algorithms which is the study of how to automatically learn from past observations to make accurate predictions and decisions. The incorporation of machine learning algorithms will bring intelligence into various kinds of systems and enable them to handle uncertainty in the interaction of physical world. In below, we will briefly discuss the motivations of different types of machine learning problems in the context of CPS applications:

- Feature representation in CPS

The wide use of high-resolution sensors leads to high-dimensional data sets in many CPS. The learning performance will be reduced when the dimensionality increases for limited labeled data, which is usually referred to as the “curse of dimensionality” issue. Instead of using all features, it is better to select those most important features to reduce redundancy and irrelevancy among data, thereby improving the learning performance and reducing the computational burden as well. Most existing feature selection methods consider the dependency and relevancy among features as the rule, and their theoretical analysis is missing. An optimal feature selection rule is needed to select those features with maximum discriminative capacity for discrimination. In Chapter 2, we will discuss this optimal feature selection rule in detail.

- Classification in CPS

In machine learning, classification is the problem of classifying a new observation into given set of categories on the basis of a training data set (past observations whose category memberships are known). Many problems could be formulated as the classification problem which has wide applications including object recognition, speech recognition, face recognition and verification, disease prediction, DNA prediction, behavior recognition, etc. The implementation of these classification methods would bring new functions into different CPS. For example, object recognition and behavior analysis with camera sensor data will help smart transportation systems to have better understanding of surrounding environment, which is necessary for autonomous vehicles to support automated planning and control to achieve desired destinations. Disease prediction such as Parkinson's disease with patient examination records or behavioral changes will help smart healthcare to provide better healthcare to these patients. In past few decades, a large number of classification methods have been developed, which can be categorized into two types of learning: generative learning and discriminative learning. We will discuss these two new learning frameworks in detail: one is the generative learning based on distribution construction in Chapter 3, and the other one is the discriminative learning based on nearest neighbors in Chapter 4.

- Anomaly detection in CPS

Anomaly detection (or outlier detection) is another important task in machine learning and data mining, which is the identification of observations which do not follow the patterns of other observations in a data set. Anomaly detection can be widely used to protect the safety of many CPS, since many

malicious events and intrusions are usually rare and different from normal activities. For example, anomaly detection can be used to detect false data injection attacks to avoid potential blackouts in smart grid, and can be also used to detect network intrusions to avoid traffic congestion in smart network. The abnormal behavior or operation detection in smart critical infrastructures such as nuclear plants can prevent large-scale disasters. Classification and clustering are two major types of anomaly detection methods. While classification methods require a training data set in which data are labeled as “normal” and “abnormal” and train a classifier in a supervised manner, clustering methods detect anomalies in an unsupervised manner under the assumption that anomalies are different from the majority of data. In Chapter 4, we will propose a new method to detect both local and global anomalies in an unsupervised manner, and in Chapter 6, we will develop a statistical method for false data injection detection for smart grid in a supervised manner.

1.3 High-Performance Computing Architectures in CPS

To implement machine learning algorithms into CPS, we further need to investigate new computing architectures to satisfy the requirements of IoT and big data analysis. At first, since the tight coupling of computation, networking and physical processes in CPS, the communication in CPS is dominated by the M2M communication. Typically, the M2M communication refers to the communications among different sensors, computing devices, actuators, and physical terminal components in CPS, as shown in Fig. 1.1. It can be shown that the M2M communication is an essential enabler of CPS.

Meanwhile, many CPS are facing the challenge of big data analysis, since the large number of sensors are deployed and continuously generate massive data

streams. The collected sensor data are transferred through the network to computing devices in which these sensor data are processed and analyzed to support intelligent decision making. For example, in smart cities, all kinds of sensors are needed to measure environmental changes such as temperature, pressure, and humidity, and to protect the safety of critical infrastructure components. In autonomous vehicles, cameras, lasers and radars are necessary components to sense the surrounding environment to detect possible collisions. While the above mentioned machine learning algorithms can provide solutions of effective data analysis including feature representation, information extraction and decision making, it also requires high-performance computing to support these data analysis over multiple spatial and temporal scales.

The design of tightly-coupled CPS poses new challenges related to distributed computing, timing requirement, failures, and data communications. Cloud computing is commonly considered as the high-performance distributed computing. The integration of CPS and Cloud computing, called Cyber-Physical Cloud Computing (CPCC), could be a solution, where all data flows get into data centers, are stored with distributed file systems, and are processed with distributed computing technologies. However, CPCC may fail to offer real-time decision making. For many critical applications, it would be too late to receive the centralized commands from data centers. To support both IoT and big data analysis, we will investigate a novel hierarchical distributed Fog computing architecture in Chapter 5.

1.4 Smart-X applications

The integration of computational intelligence brings “smartness” into various kinds of physical systems. For example, smart vehicles with object recognition and behavior analysis will have better understanding of surrounding environment and be able to avoid potential collisions with obstacles and pedestrians when driving on

the road; smart grid with attacks detection will lead to a safer energy system; smart farm with crop identification will enable agriculture automation; smart building with state prediction will achieve optimal operations and save a lot of energy in future buildings. All these intelligent CPS could be considered as the class of system of Smart-X.

Notice that different Smart-X systems have their unique computational applications with different types of sensor data. In this thesis, we will investigate three Smart-X applications: smart pipeline, robotics, and smart grid. Particularly, we apply intelligent learning algorithms to physical systems to achieve desired functionalities. In Chapter 5, as a prototype of the proposed Fog computing architecture, we will apply feature reduction and spatial-temporal learning method to identify hazardous events to protect the safety of pipelines (gas/oil/water). In Chapter 6, the detection of false data injection will be discussed in detail to protect the security of smart grid.

1.5 Summary of Thesis Contributions

The contributions of this thesis are three parts towards intelligent CPS: machine learning algorithms, computing architectures, and applications. For the part of machine learning algorithms, we contribute one optimal feature selection framework to further facilitate the learning for high-dimensional data and two novel learning frameworks: EEF-based generative learning and ENN-based discriminative learning, for classification and anomaly detection in many CPS. For the part of computing architecture, we propose a hierarchical distributed Fog computing architecture to support both big data analysis and IoT. A prototype of this architecture is also developed to demonstrate its feasibility. For the applications of CPS, we investigate the integration of computational intelligence into power systems. The brief summary of the key contributions of this thesis is outlined as

follows:

Machine Learning Algorithms:

1. *Toward Optimal Feature Selection for Classification.* Automated feature selection is important for classification to reduce feature size and to speed up the learning process of classifiers. A novel and efficient feature selection framework based on the Information Theory is proposed, which aims to rank the features with their discriminative capacity. A new divergence measure, called Jeffreys-Multi-Hypothesis (JMH) divergence, is introduced to measure multi-distribution divergence for a general classification problem. Based on the JMH-divergence, two efficient feature selection methods, termed maximum discrimination (MD) and $MD - \chi^2$ methods, are developed to select the optimal feature subsets.
2. *EEF: Exponentially Embedded Family for Classification.* EEF is a novel probability density function (PDF) construction method based on known reference distributions. Given a set of training data, a new generalized PDF is constructed with respect to the reference distribution. Based on the constructed PDF, a decision rule is built for classification in either a data-driven manner or a model-driven manner. The proposed method provides an effective way to address the problem of model parameters learning using embedded latent variable exponential family probability for classification.
3. *GPPT: Generalized PDF Projection Theorem via EEF.* Information loss may always happen in feature reduction, which will reduce the learning performance. To address this information loss issue in feature reduction, a novel exponentially embedded families (EEF) based classification method is proposed, in which the probability density function (PDF) on raw data is estimated from the PDF on features. With the PDF construction, class-specific

features can be used in the proposed classification method, instead of a common feature subset for all classes as used in conventional approaches.

4. *ENN: Extended Nearest Neighbor Method for Classification.* A new supervised classification method, extended nearest neighbor (ENN), is proposed according to the maximum gain of intra-class coherence. Unlike the classic k -nearest neighbor (KNN) method, in which only the nearest neighbors of a test sample are used to estimate a group membership, the ENN method makes a prediction in a “two-way communication” style: it considers not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors. By exploiting the generalized class-wise statistics from all training data by iteratively assuming all the possible class memberships of a test sample, the ENN is able to learn from the global distribution, therefore improving pattern recognition performance and providing a powerful technique for a wide range of data analysis applications.

5. *A Local Density-Based Approach with Extended Nearest Neighbors for Local Outlier Detection.* Outlier detection is widely used in many CPS to detect possible malicious behaviors and bad data. One simple but effective density-based outlier detection approach with the local kernel density estimation (KDE) is developed. A Relative Density-based Outlier Score (RDOS) is introduced to measure the local outlierness of objects. Instead of using only k nearest neighbors, both reverse nearest neighbors and shared nearest neighbors of an object are used for density distribution estimation. Some theoretical properties of the RDOS including its expected value and false alarm probability are derived to help select important parameters for outlier detection in real systems.

Computing Architectures:

6. *A Hierarchical Distributed Fog Computing Architecture in CPS.* The natural characteristic of geo-distribution in CPS requires a new computing paradigm to offer location-awareness and latency-sensitive monitoring and intelligent control. Fog Computing, which extends the computing to the edge of network, fits this need. A hierarchical distributed Fog Computing architecture is proposed to support the integration of massive number of infrastructure components and computations. A working prototype is also constructed to experimentally evaluate the feasibility of the proposed architecture for smart pipeline monitoring.

Applications:

7. *False Data Injection Detection in Smart Grid.* In smart energy systems, state estimation and false data injection detection are two important aspects to maintain normal operations. This work considers the situation when the measurements are corrupted by colored Gaussian noise. By modeling the noise with the autoregressive process, the states of the power transmission networks are estimated and a generalized likelihood ratio test (GLRT) detector is developed to detect false data injection attacks. It can be shown that the conventional approach with the assumption of Gaussian noise is a special case of the proposed method, and thus the new approach has more applicability.

1.6 Related Work

The general machine learning algorithms and computing architectures proposed in this thesis could be used in various CPS. In the section, we give a brief literature review of topics in each chapter. Closely related work will be discussed

in detail in later chapters.

Effective feature representation is an important topic in the learning from high-dimensional data. Feature selection methods discussed in Chapter 2 aim to select those most important features to facilitate the learning. In past decades, a number of feature selection methods have been proposed, which can be usually categorized into the following two types of approaches: the *filter* approach and the *wrapper* approach [1]. The filter approach selects feature subsets based on the general characteristics of the data without involving the learning algorithms that will use the selected features. A score indicating the “importance” of the feature is assigned to each individual feature based on an independent evaluation criterion [2, 3, 4, 5], such as distance measure, entropy measure, dependency measure and consistency measure. After that, the filter approach only selects a number of top ranked features and ignores the rest. Alternatively, the wrapper approach greedily searches for better features with an evaluation criterion based on the same learning algorithm [6][7]. Although it has been shown that the wrapper approach usually performs better than the filter approach, it has much more computational cost than the filter approach, which sometimes makes it impractical. Typically, the filter approach is predominantly used because of its simplicity and efficiency. However, the filter approach evaluates the goodness of a feature by only exploiting the intrinsic characteristics of the training data without considering the learning algorithm for discrimination, which may lead to an undesired classification performance. Given a specific learning algorithm, it is difficult to determine which filter feature selection approach is the best for discrimination in a theoretical way. In Chapter 2, we develop an optimal feature selection framework based on information theory to select those features with maximum discriminative capacity to achieve best performance.

Supervised learning algorithms fall into one of two categories: generative learning and discriminative learning [8]. While the discriminative learning methods learn a mapping function from input variables to output class labels, the generative learning methods learn underlying joint distribution models of the given data. The EEF-based learning in Chapter 3 is a type of generative learning which is closely related to the learning with PDF estimation [9][10][11]. The PDF estimation is a long-standing research area in both statistics and machine learning. Traditionally, two types of PDF estimation methods have been widely used: *parametric* and *nonparametric* methods, which lead to parametric and non-parametric classifiers, respectively. The parametric methods assume that the underlying joint distributions/models of the data are known except for the “source” parameters of their PDFs which need to be estimated [8], such as naive Bayes classifier with Gaussian distribution models or multinomial distribution models. In contrast, the non-parametric methods do not rely on the assumption of data distributions such as Gaussian mixture models and kernel density estimation (KDE) methods [12] [13] [14] [15]. The non-parametric approaches offer a greater flexibility in modeling a given dataset, and have been successfully applied in applications like classification [15] [16], discriminative inference [17], background modeling [18], deformable shape and appearance modeling [19] and object tracking in video analysis [20]. In Chapter 3, we consider a new PDF estimation method based on EEF, and build a new class of classifiers in either parametric manner or non-parametric manner.

The ENN-based learning in Chapter 4 is a type of discriminative learning, which is motivated by the limitations of traditional KNN method. KNN has wide applications in data mining due to its easy implementation, competitive performance, and nonparametric computational basis which is independent of the underlying data distribution. The first modern study of KNN can be traced back to 1951

by Fix and Hodges [21]. In their formalization of nonparametric discrimination, the consistency of KNN was established using a probability density estimation. In another representative work [22], Cover and Hart proved that the classification error of 1NN ($K=1$) is at most twice Bayes optimal error. These early classic works laid down a strong theoretical foundation for KNN-based methods, which have been witnessed in considerable applications in many different disciplines, such as biological and chemical data analysis [23], disease classification and clinical outcome prediction [24], among others. However, KNN is sensitive to the scale or variance of the distributions of the predefined classes [25], which leads to the “two types of errors” [26]. To address this issue, we propose an ENN method and its variation for classification and anomaly detection in Chapter 4.

High-performance computing architecture in Chapter 5 is an essential enabler of big data analysis in CPS. The traditional “pay-as-you-go” Cloud computing paradigm could be a possible solution. The integration of CPS and Cloud computing, called Cyber-Physical Cloud Computing (CPCC) [27, 28, 29], has been studied in CPS. However, in CPCC, all data flows will get into data centers, be stored with distributed file systems, and be processed with distributed computing technologies. It would be very difficult to achieve real-time decision making over multiple spatial and temporal scales. Meanwhile, many CPS require to support IoT such that M2M communications could be dominated. To support both big data analysis and IoT, Fog computing [30] is a emerging computing architecture and hot topic in CPS. In Chapter 5, a hierarchical distributed Fog computing architecture is proposed and its feasibility is demonstrated by implementing a prototype for smart pipeline.

The CPS application studied in Chapter 6 is to protect the safety of smart grid through state estimation and false data injection (FDI) attacks. Recent studies

have found that malicious attackers can inject false data into the measurements which will threaten the normal operations of energy systems [31, 32]. Defense studies have also been taken to investigate the protection, detection, and mitigation based on specific features of corresponding attack schemes [33, 34, 35]. In Chapter 6, a detectability analysis of false data injection is given and a novel FDI detection method is developed when the measurements are corrupted by colored Gaussian noise.

1.7 Thesis Organization

The organization of this thesis is shown in Fig. 1.2. It includes three parts of CPS: algorithms, architectures and applications. We assume that all kinds of sensors have been already deployed in physical components and the sensor data have been collected. With the collected sensor data, we focus on the development of advanced machine learning algorithms to bring intelligence into CPS. These machine learning algorithms need to be implemented into a high-performance and distributed computing architecture for various CPS applications.

Chapter 2 provides a new feature selection framework to select those optimal feature subsets for classification. Chapter 3 introduces the theory of EEF for distribution construction and develops classification rules based on EEF in an either data-driven manner or model-driven manner. Chapter 4 develops a new concept of ENN for classification and anomaly detection, with their theoretical analysis and performance evaluation for real-life applications. Chapter 5 presents a hierarchical distributed Fog computing architecture in CPS and implements its prototype for smart pipeline. Chapter 6 studies one important CPS application in smart grid with the incorporation of intelligence to detect false data injection attacks. Chapter 7 summarizes the contributions of this thesis and discusses some future research directions.

Physical Components

Algorithms

Architectures

Applications

Figure 1.2: Thesis organization.

CHAPTER 2

Feature Representation

In many CPS, a large amount of sensors are deployed to measure system states and environmental changes, leading to high-dimensional sensor data. The limited labeled data would become sparse as the dimension increases. This sparsity would hurt the learning performance for any machine learning algorithms. It is necessary to reduce feature dimensions before training a learning algorithm, which is also called feature reduction. *Feature selection* which only keeps those most important features is a very important type of feature reduction method because of its simplicity and efficiency. This chapter starts by giving an introduction of feature selection in §2.1, and then describes a novel theoretical framework toward the optimal feature selection in §2.2. In §2.3, several efficient feature selection algorithms are proposed, taking text categorization as an example. §2.4 shows the improved prediction performance on several benchmarks compared with the state-of-the-art feature selection methods.

2.1 Introduction

The “curse of dimensionality” is a critical issue in the learning from high-dimensional data sets. With the limited training data, the irrelevant and redundant features may reduce predictive performance for any machine learning algorithms. Meanwhile, tens and hundreds of thousands features may lead to a high computational burden for the learning process. To avoid the issue of the “curse of dimensionality” and to speed up the learning process, it is necessary to perform feature reduction to reduce feature size.

Feature selection is a common feature reduction approach for many learning algorithms, where only a subset of original features are selected as input to the

learning algorithms. In past decades, a number of feature selection methods have been proposed, which can be mainly categorized into the following two types of approach: the *filter* approach and the *wrapper* approach [1]. The filter approach selects feature subsets based on the general characteristics of the data without involving the learning algorithms that will use the selected features. A score indicating the “importance” of the term is assigned to each individual feature based on an independent evaluation criterion, such as distance measure, entropy measure, dependency measure and consistency measure. Hence, the filter approach only selects a number of top ranked features and ignores the rest. Alternatively, the wrapper approach greedily searches for better features with an evaluation criterion based on the same learning algorithm. Although it has been shown that the wrapper approach usually performs better than the filter approach, it has much more computational cost than the filter approach, which sometimes makes it impractical.

Typically, the filter approach is predominantly used because of its simplicity and efficiency. However, the filter approach evaluates the goodness of a feature by only exploiting the intrinsic characteristics of the training data without considering the learning algorithm for discrimination, which may lead to an undesired classification performance. Given a specific learning algorithm, it is hard to determine which filter feature selection approach is the best for discrimination.

In this chapter, we first present a feature selection method which ranks the original features, aiming to maximize the discriminative performance for classification, when naive Bayes classifiers are used as learning algorithms. Unlike those existing filter approaches, our method evaluates the goodness of a feature without training a classifier explicitly, and selects these features that offer maximum discrimination in terms of a new divergence measure. However, information loss may always happen in feature reduction as some features are discarded. To avoid the

information loss issue, we further develop a new learning algorithm to project the distribution from feature subspace back into raw data space and make classification decisions in raw data space. Throughout this chapter, we take text categorization as an example and evaluate the performance of the proposed methods for several well-known benchmarks for text categorization.

2.2 Theoretical Framework of Optimal Feature Selection

We follow the Information Theory to select feature subsets that have maximum discriminative capacity for distinguishing the samples among two or more classes. We first introduce some concepts on information measures for binary hypothesis testing (also known as “two-class” classification) and present a new divergence measure for multiple hypothesis testing (i.e., for “multi-class” classification).

2.2.1 Divergence Measures for Binary Hypothesis Testing

Considering a two-class classification problem first, each class is represented by a particular distribution, saying $P_1 = p(x|c_1; \theta_1)$ for class c_1 and $P_2 = p(x|c_2; \theta_2)$ for class c_2 . A test procedure for classification can be considered as a binary hypothesis testing such that if a sample is drawn from P_1 we accept the hypothesis H_1 (reject the hypothesis H_2), and if a sample is drawn from P_2 we accept H_2 (reject H_1). In other words, we have $p(x|c_1) = p(x|H_1)$ and $p(x|c_2) = p(x|H_2)$, and we also denote $p(x|H_i)$ as the class conditional probability distribution in the rest of chapter.

According to the Information Theory [36][37], we define the KL-divergence $\mathcal{KL}(P_1, P_2)$ between two probability distributions (from P_1 to P_2) as

$$\begin{aligned}\mathcal{KL}(P_1, P_2) &= \int_x p(x|H_1) \log \frac{p(x|H_1)}{p(x|H_2)} dx \\ &= E_{p_1} \left[\log \frac{p(x|H_1)}{p(x|H_2)} \right]\end{aligned}\tag{2.1}$$

where $E_{p_1}[x]$ denotes the expectation of x with respect to the probability distribu-

tion P_1 .

Under the MAP rule in Eq. (2.18), we would classify the sample x into class c_1 , i.e., accept H_1 , if

$$\begin{aligned} \log p(x|H_1) + \log p(H_1) &> \log p(x|H_2) + \log p(H_2) \\ \Rightarrow \log \frac{p(x|H_1)}{p(x|H_2)} &> -\log \frac{p(H_1)}{p(H_2)} = \gamma \end{aligned} \quad (2.2)$$

The logarithm of the likelihood ratio, $\log \frac{p(x|H_1)}{p(x|H_2)}$, measures the information of the observation x for discrimination in favor of H_1 against H_2 [36]. From the definition of the KL-divergence measure in Eq. (2.1), then $\mathcal{KL}(P_1, P_2)$ indicates the mean information for discrimination in favor of H_1 against H_2 .

Taking the expectation with respect to the distribution P_1 in Eq. (2.2) (i.e., considering all possible observations drawn from P_1), we have

$$\mathcal{KL}(P_1, P_2) > \gamma \quad (2.3)$$

which also illustrates that $\mathcal{KL}(P_1, P_2)$ is an indicator of a Bayesian classifier's discriminative capacity for discriminating the observation from class c_1 in favor of H_1 against H_2 . With the extension of the central limit theorem, Chernoff in [38] showed that, for a large number of observations, the type I error α^* , the probability of incorrectly accepting H_1 , asymptotically has

$$\lim_{n \rightarrow \infty} \log \frac{1}{\alpha^*} = \lim_{n \rightarrow \infty} \mathcal{KL}(P_1, P_2; O_n) \quad (2.4)$$

where O_n denotes n independent observations. It has been said that a larger value of KL-divergence indicates a lower type I error when there are infinite number of observations.

Note that the KL-divergence measure is not symmetric. Alternatively, the KL-divergence $\mathcal{KL}(P_2, P_1)$ indicates a Bayesian classifier's discriminative capacity for discriminating the observation from class c_2 in favor of H_2 against H_1 . Similarly,

under the MAP rule in Eq. (2.18), considering all the observation drawn from class c_2 , we have

$$\mathcal{KL}(P_2, P_1) > -\gamma \quad (2.5)$$

For a large number of observations, the type II error β^* , the probability of incorrectly accepting H_2 , can be given by

$$\lim_{n \rightarrow \infty} \log \frac{1}{\beta^*} = \lim_{n \rightarrow \infty} \mathcal{KL}(P_2, P_1; O_n) \quad (2.6)$$

To minimize both type I error and type II errors in an asymptotic way, the Jeffreys divergence (J-divergence) can be used, which is defined by [39]

$$\mathcal{J}(P_1, P_2) = \mathcal{KL}(P_1, P_2) + \mathcal{KL}(P_2, P_1) \quad (2.7)$$

By combining Eq. (2.3) and Eq. (2.5), we have

$$\mathcal{KL}(P_1, P_2) > \gamma > -\mathcal{KL}(P_2, P_1) \quad (2.8)$$

Because

$$\mathcal{KL}(P_1, P_2) \geq 0, \quad \mathcal{KL}(P_2, P_1) \geq 0 \quad (2.9)$$

a larger $\mathcal{J}(P_1, P_2)$ may lead to a smaller type I and type II error asymptotically. It is known that the J-divergence is able to measure the difficulty and capacity of discriminating between H_1 and H_2 [36] [38] [39].

The purpose of feature selection is to determine the most informative features which lead to the best prediction performance. Hence, it is natural to select those features that have the maximum discriminative capacity for classification, by minimizing the classification error (i.e., maximizing the KL-divergence or the J-divergence). However, the J-divergence is only defined for binary hypothesis. We next extend the J-divergence for multiple hypothesis testing (i.e., multi-class classification).

2.2.2 Jeffreys-Multi-Hypothesis Divergence

The Jensen-Shannon (JS) divergence [40] is the one that can be used to measure multi-distribution divergence, in which the divergences of each individual distribution with a reference distribution are calculated and summed together. Unlike the J-divergence, the measure of discrimination capacity may not hold. In [41], Sawyer presents a variant of J-divergence with a variance-covariance matrix for multiple comparisons of separate hypotheses. Here, we first generalize the J-divergence to a multi-distribution using the scheme of “one-vs-all” [42], which is defined as follows:

Definition 1. Let $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$ be the set of N distributions. The Jeffreys-Multi-Hypothesis (JMH) divergence, denoted by $\mathcal{JMH}(P_1, P_2, \dots, P_N)$, is defined to be

$$\mathcal{JMH}(P_1, P_2, \dots, P_N) = \sum_{i=1}^N \mathcal{KL}(P_i, \bar{P}_i) \quad (2.10)$$

where $\bar{P}_i$ is the combination of all remaining $N - 1$ distributions $\bar{P}_i = \sum_{k=1, k \neq i}^N \pi_{ki} P_k$, and π_{ki} are the prior coefficients.

Similar to the “one-vs-all” strategy for multi-class classification problem, we build N binary hypothesis testing detectors, each of which discriminates the samples in favor of P_i against $\bar{P}_i$ which is the complement of P_i . In each detector, $\bar{P}_i$ is represented by a mixture distribution over all the remaining $N - 1$ classes with the coefficients π_{ki} which are given by

$$\pi_{ki} = \frac{p_{c_k}}{\sum_{m=1, m \neq i}^N p_{c_m}} \quad (2.11)$$

where p_{c_m} is the prior probability of class c_m . Since the KL-divergence of each detector is the measure of its discriminative capacity for discrimination, the new multi-distribution divergence is able to measure the discrimination capacity over all classes. Specifically, when $N = 2$, we have $\mathcal{JMH}(P_1, P_2) = \mathcal{J}(P_1, P_2)$.

Note that, since the JMH divergence is the sum of multiple J-divergences, it holds most properties of J-divergence. For example, JMH divergence is almost positive definite, i.e., $\mathcal{JMH}(P_1, P_2, \dots, P_N) \geq 0$, with equality if and only if $p_{1m} = p_{2m} = \dots = p_{Nm}$, $m = 1, 2, \dots, M$. Also it is a symmetric measure, that is, $\mathcal{JMH}(\dots, P_i, \dots, P_k, \dots) = \mathcal{JMH}(\dots, P_k, \dots, P_i, \dots)$.

2.2.3 Optimal Feature Subset Selection

Unlike those existing feature selection methods which compute the score (“importance”) of features based on the feature relevance to class, our goal is to select the features that offer the maximum discrimination for classification. By doing so, one can expect an improved classification performance.

For a two-class classification problem, we know that the J-divergence indicates the discriminative capacity of discriminating two classes data under the MAP rule. Hence, we can formulate the feature selection problem as follows: given a set of M features $\mathcal{B}$ where $|\mathcal{B}| = M$ and a predetermined integer r , the number of features to be selected, we aim to find the most r discriminative features $\mathcal{B}_r^* \subset \mathcal{B}$ where $|\mathcal{B}^*| = r$, such that,

$$\begin{aligned} \mathcal{B}_r^* &= \arg \max_{\mathcal{B}_r \subset \mathcal{B}} \mathcal{J}(P_1, P_2 | \mathcal{B}_r) \\ &= \arg \max_{\mathcal{B}_r \subset \mathcal{B}} \mathcal{KL}(P_1, P_2 | \mathcal{B}_r) + \mathcal{KL}(P_2, P_1 | \mathcal{B}_r) \end{aligned} \quad (2.12)$$

where $\mathcal{J}(\cdot, \cdot | \mathcal{B}_r)$ and $\mathcal{KL}(\cdot, \cdot | \mathcal{B}_r)$ are the J-divergence and KL-divergence, respectively, when a subset of features $\mathcal{B}_r \subset \mathcal{B}$ are considered. For a general multi-class classification problem, the JMH-divergence is needed to replace the J-divergence in Eq. (2.12).

Considering an N -class classification problem, the r most discriminative fea-

tures $\mathcal{B}_r^*$ are selected by maximizing the JMH-divergence, which are given by

$$\mathcal{B}_r^* = \arg \max_{\mathcal{S}_r \subset \mathcal{B}, |\mathcal{S}_r|=r} \mathcal{JMH}(P_1, P_2, \dots, P_N | \mathcal{S}_r) = \arg \max_{\mathcal{S}_r \subset \mathcal{B}, |\mathcal{S}_r|=r} \sum_{i=1}^N \mathcal{KL}(P_i, \bar{P}_i | \mathcal{S}_r) \quad (2.13)$$

where $\mathcal{JMH}(P_1, P_2, \dots, P_N | \mathcal{S}_r)$ is the JMH-divergence defined in Eq. (2.10) with the feature subset $\mathcal{S}_r$. Note that each KL-divergence indicates the discriminative capacity of one binary classifier to distinguish the samples in one class P_i from the samples in all remaining classes $\bar{P}_i$, and thus the JMH-divergence is able to measure the difficulty and capacity of discriminating the samples among all classes.

The problems in Eq. (2.12) and Eq. (2.13) are also known as the NP-hard problems, if one examines each of $\binom{M}{r}$ combinations, which is intractable particularly for a high dimensional data set. Meanwhile, in practice, we need to examine various r values to evaluate the classification performance using those selected r features. Hence, it is necessary to seek efficient algorithms, e.g., assigning an importance score to each feature and ranking the features. In the sequel, we take text categorization as an example and develop several efficient algorithms to select near optimal feature subsets.

2.3 Efficient Optimal Feature Subsets Selection for Text Categorization

2.3.1 Background of Text Categorization

Text categorization is a task of assigning a topic to a document and has important applications in text mining. Each document in text categorization is commonly represented by the model of the “bag-of-words” with a $M \times 1$ feature vector $D : \mathbf{d} = [x_1, x_2, \dots, x_M]^T$. The i -th element x_i corresponds to the measure of the i -th term (“word”) in a vocabulary or a dictionary. For a given data set, we first generate the vocabulary with a set of M unique terms from all documents. Then, for each document, a feature vector can be formed by using various feature models. Typically, the value of a feature represents the information about this

particular term in a document.

MNB would be one of the best-known naive Bayes classification approaches using the term frequency to represent the document. Considering a text categorization problem with N classes (“topics”), let c be the discrete variable of class label taking values in $\{1, 2, \dots, N\}$, and $\mathbf{x}$ be the integer-valued feature vector corresponding to the term frequency. The MNB classifier assumes that the number of times that each term appears in the document satisfies a multinomial distribution [43][44]. In other words, a document with l terms is considered as l independent trials, and each term is the result of a trial exactly falling into the vocabulary. Let the vocabulary size be M , and then each documents is represented by a $M \times 1$ feature vector.

Hence, given a document D , we first count the number of times that each term appears and generate a feature vector $\mathbf{d} = [x_1, x_2, \dots, x_M]^T$. According to the multinomial distribution, the likelihood of observing $\mathbf{d}$ conditioned on the class label c and the document length l can be calculated as follows:

$$p(\mathbf{d}|c, l; \boldsymbol{\theta}_c^m) = \frac{l}{x_1!x_2!\dots x_M!} \prod_{i=1}^M p_{ic}^{x_i} \quad (2.14)$$

where $l = \sum_{m=1}^M x_m$, and $\boldsymbol{\theta}_c^m = [p_{1c}, p_{2c}, \dots, p_{Mc}]^T$ is a parameter vector, each of which denotes the probability that the term of a trial falls into one of M categories, so that $p_{ic} \geq 0$ for $i = 1, 2, \dots, M$ and $p_{1c} + p_{2c} + \dots + p_{Mc} = 1$. Using the MLE method, the estimate of each term probability $\hat{p}_{ic}$ is given by

$$\hat{p}_{ic} = \frac{l_{ic}}{l_c} \quad (2.15)$$

where l_{ic} is the number of times the i -th term appears among documents in class c , and l_c is the total number of terms in class c . To avoid the *zero probability* issue, the technique of “Laplace smoothing” or the prior information for probability p_{ic}

is usually applied [45], which leads to

$$\hat{p}_{ic} = \frac{l_{ic} + \beta_1}{l_c + \beta_2} \quad (2.16)$$

where β_1 and β_2 are the constant smoothing parameters. Using the Laplacean prior, we prime each word's count with a count of one and have $\beta_1 = 1$ and $\beta_2 = M$ [44][46].

Note that the document length l is commonly assumed to be independent on the document class to simplify the naive Bayes classification rule, that is, we have $p(\mathbf{d}|c, l) = p(\mathbf{d}|c)$ and $p(c|\mathbf{d}, l) = p(c|\mathbf{d})$ for likelihood and posterior probability, respectively. Otherwise, it leads to a more general formulation of the posterior probability for classification, which is given by

$$p(c|\mathbf{d}, l) = p(\mathbf{d}, l|c)p(c) = p(\mathbf{d}|c, l)p(l|c)p(c) \quad (2.17)$$

where the class-wise document length information $p(l|c)$ is incorporated in a Bayesian fashion. The document length information sometimes may be useful for making classification decisions, e.g., when the class-wise document length distributions are different, but it requires the estimate of $p(l|c)$ for a given data set. We will follow the common assumption that the document length is constant in our experiments, i.e., $p(\mathbf{d}|c) = p(\mathbf{d}|c, l)$, and note that the solutions of our feature selection methods do not change with and without this assumption.

In naive Bayes classifiers, given a new document D_t to be classified, we first generate its feature vector $\mathbf{d}_t$ and apply the following decision rule to make a classification:

$$\begin{aligned} c^* &= \arg \max_{c \in \{1, 2, \dots, N\}} p(c|\mathbf{d}_t; \boldsymbol{\theta}_c) \\ &\propto \arg \max_{c \in \{1, 2, \dots, N\}} p(\mathbf{d}_t|c; \boldsymbol{\theta}_c)p(c) \\ &\propto \arg \max_{c \in \{1, 2, \dots, N\}} \log p(\mathbf{d}_t|c; \boldsymbol{\theta}_c) + \log p(c) \end{aligned} \quad (2.18)$$

The likelihood probability $p(\mathbf{d}_t|c; \boldsymbol{\theta}_c)$ could be a specific model-based distribution. Here, for MNB classifier, the probability distributions $p(\mathbf{d}_t|c; \boldsymbol{\theta}_c)$ for $i = 1, 2, \dots, N$, are the multinomial distributions given by Eq. (2.14).

Meanwhile, it is easy to obtain the KL-divergence measure in Eq. 2.1 between two MNB distributions. According to Eq. (2.14) and Eq. (2.1), the KL-divergence between two multinomial distributions P_1^m and P_2^m is given by

$$\mathcal{KL}(P_1^m, P_2^m) = \sum_{i=1}^M p_{i1} \log \frac{p_{i1}}{p_{i2}} \quad (2.19)$$

where $0 \leq p_{ic} \leq 1$ and $\sum_{i=1}^M p_{ic} = 1$ for $i = 1, 2, \dots, M$, $c = 1, 2$. It can be shown that we can obtain a closed form of J-divergence and JMH-divergence in Eq. 2.12 for optimal feature subsets selection.

2.3.2 A Greedy Feature Selection Approach

We consider a binary (two-class) classification problem first and extend our feature selection method to a general multi-class classification problem later. Here, we start to propose a greedy approach to rank the features according to their discriminative capacity for naive Bayes. This approach starts to determine which feature of the M features produces the maximum JMH-divergence if only one single feature is used for classification. To determine the most discriminative feature, for each feature x_i , $i = 1, 2, \dots, M$, we build two variables: x_i and $\bar{x}_i$, where x_i is the original i -th feature and $\bar{x}_i$ is the pool of all the remaining $M - 1$ features with parameters $\bar{p}_{ic}^k$. We use the superscript number k to indicate the k -th step of our greedy approach (e.g., $k = 1$ here). We have

$$\begin{aligned} \bar{p}_{i1}^1 &= \sum_{m=1, m \neq i}^M p_{m1} = 1 - p_{i1} \\ \bar{p}_{i2}^1 &= \sum_{m=1, m \neq i}^M p_{m2} = 1 - p_{i2} \end{aligned} \quad (2.20)$$

We denote the distributions of these two variables for class 1 and class 2 by P_{i1}^1 and P_{i2}^1 , respectively. Note that P_{i1}^1 and P_{i2}^1 also satisfy multinomial distribution but with different parameters. Then, we calculate the J-divergence between P_{i1}^1 and P_{i2}^1 as $\mathcal{J}_i^1(P_{i1}^1, P_{i2}^1) = \mathcal{KL}(P_{i1}^1, P_{i2}^1) + \mathcal{KL}(P_{i2}^1, P_{i1}^1)$ with Eq. (2.19). At the end of the 1st step, we obtain M J-divergences $\mathcal{J}_i^1(P_{i1}^1, P_{i2}^1)$ for $i = 1, 2, \dots, M$, and choose the first feature indexed by s_1 that leads to the maximum J-divergence, that is,

$$s_1 = \arg \max_{i=1,2,\dots,M} J_i^1(P_{i1}^1, P_{i2}^1) \quad (2.21)$$

Then, we fix the first feature s_1 and repeat the process over the remaining features. Specifically, at the k -th step to select the k -th feature, let $\mathcal{S}_{k-1} = \{s_1, s_2, \dots, s_{k-1}\}$ be the feature index set which are selected from the previous $k-1$ steps. Again, for each individual feature x_i , $i = 1, 2, \dots, M$, $i \notin \mathcal{S}_{k-1}$, we form $k+1$ variables: $x_{s_1}, x_{s_2}, \dots, x_{s_{k-1}}, x_i, \bar{x}_i$, where the first k variables are the original features and the last variable $\bar{x}_i$ is the pool of all remaining $M-k$ features with parameters $\bar{p}_{ic}^k$ for two classes ($c = 1, 2$). We have

$$\begin{aligned} \bar{p}_{i1}^k &= 1 - p_{i1} - \sum_{m \in \mathcal{S}_{k-1}} p_{m1} \\ \bar{p}_{i2}^k &= 1 - p_{i2} - \sum_{m \in \mathcal{S}_{k-1}} p_{m2} \end{aligned} \quad (2.22)$$

Denote the distributions of these variables for class 1 and class 2 at the k -th step by P_{i1}^k and P_{i2}^k , respectively. At the k -th step, we choose the feature indexed by s_k with the following maximum J-divergences:

$$s_k = \arg \max_{i=1,2,\dots,M, i \notin \mathcal{S}_{k-1}} J_i^k(P_{i1}^k, P_{i2}^k) \quad (2.23)$$

Hence, at the end of the M -th step, a ranked feature index set $\mathcal{S}_M = \{s_1, s_2, \dots, s_M\}$ is produced. The implementation of this greedy feature selection

Algorithm 1: A Greedy Feature Selection Algorithm Based on the Maximum J-Divergence for Two-class Classification

INPUT:

- The estimated probabilities of M features: $\theta_i = [p_{1c}, p_{2c}, \dots, p_{Mc}]$, $c = 1, 2$ for two classes;
- The ranked feature index set: $\mathcal{S} = \emptyset$.
- The full index set: $\mathcal{S}_a = \{1, 2, \dots, M\}$

ALGORITHM:

for $k = 1 : M$ **do**

foreach $i \in \mathcal{S}_a \cap \bar{\mathcal{S}}$ **do**

1. Form $k + 1$ variables: $x_{s_1}, x_{s_2}, \dots, x_{s_{k-1}}, x_i$ and $\bar{x}_i$, and denote P_{i1}^k and P_{i2}^k for class 1 and 2 distribution, respectively;

2. Calculate J-divergence $J_i^k(P_{i1}^k, P_{i2}^k)$ between P_{i1}^k and P_{i2}^k ;

end

3. s_k : select the feature using Eq. (2.23);

4. $\mathcal{S} = \mathcal{S} \cup s_k$;

end

OUTPUT:

- A ranked feature index set: $\mathcal{S} = \{s_1, s_2, \dots, s_M\}$.
-

approach based on the maximum J-divergence for two-class classification is given in Algorithm 1.

Theorem 1. *The maximum J-divergences $\mathcal{J}_{s_k}$ for $k = 1, 2, \dots, M$ in Algorithm 1 monotonically increases, i.e.,*

$$\mathcal{J}_{s_{k+1}}^{k+1}(P_{s_{k+1}1}^{k+1}, P_{s_{k+1}2}^{k+1}) \geq \mathcal{J}_{s_k}^k(P_{s_k1}^k, P_{s_k2}^k) \quad (2.24)$$

Proof. The J-divergence between the distribution P_1 specified by the parameters $[p_{11}, p_{21}, \dots, p_{M1}]$ for class 1 and the distribution P_2 specified by the parameters $[p_{12}, p_{22}, \dots, p_{M2}]$ for class 2 is given by

$$\mathcal{J}(P_1, P_2) = \sum_{i=1}^M p_{i1} \log \frac{p_{i1}}{p_{i2}} + \sum_{i=1}^M p_{i2} \log \frac{p_{i2}}{p_{i1}} \quad (2.25)$$

Hence, at the $(k+1)$ -th step, the maximum J-divergence $\mathcal{J}_{s_{k+1}}(P_{s_{k+1}1}^{k+1}, P_{s_{k+1}2}^{k+1})$ shorted as $\mathcal{J}_{s_{k+1}}$ is given by

$$\begin{aligned} \mathcal{J}_{s_{k+1}} = & \underbrace{\sum_{i=1}^{k+1} p_{s_i1} \log \frac{p_{s_i1}}{p_{s_i2}} + \left(1 - \sum_{i=1}^{k+1} p_{s_i1}\right) \log \frac{1 - \sum_{i=1}^{k+1} p_{s_i1}}{1 - \sum_{i=1}^{k+1} p_{s_i2}}}_{B_{12}^{k+1}} \\ & + \underbrace{\sum_{i=1}^{k+1} p_{s_i2} \log \frac{p_{s_i2}}{p_{s_i1}} + \left(1 - \sum_{i=1}^{k+1} p_{s_i2}\right) \log \frac{1 - \sum_{i=1}^{k+1} p_{s_i2}}{1 - \sum_{i=1}^{k+1} p_{s_i1}}}_{B_{21}^{k+1}} \end{aligned} \quad (2.26)$$

Following Jensen's inequality, given a set $E \subset \mathcal{R}$, we have

$$\int_E f_1(x) \log \frac{f_1(x)}{f_2(x)} dx \geq \int_E f_1(x) dx \log \frac{\int_E f_1(x) dx}{\int_E f_2(x) dx} \quad (2.27)$$

which gives

$$\begin{aligned} & p_{11} \log \frac{p_{11}}{p_{12}} + p_{21} \log \frac{p_{21}}{p_{22}} + \cdots + p_{n1} \log \frac{p_{n1}}{p_{n2}} \\ & \geq (p_{11} + p_{21} + \cdots + p_{n1}) \log \frac{p_{11} + p_{21} + \cdots + p_{n1}}{p_{12} + p_{22} + \cdots + p_{n2}} \end{aligned} \quad (2.28)$$

for $p_{ic} > 0$, $i = 1, 2, \dots, n$, $c = 1, 2$, with equality if and only if

$$\frac{p_{11}}{p_{12}} = \frac{p_{21}}{p_{22}} = \cdots = \frac{p_{n1}}{p_{n2}} = \frac{p_{11} + p_{21} + \cdots + p_{n1}}{p_{12} + p_{22} + \cdots + p_{n2}} \quad (2.29)$$

Using Eq. (2.28), we have

$$B_{12}^{k+1} \geq \sum_{i=1}^k p_{s_i1} \log \frac{p_{s_i1}}{p_{s_i2}} + \left(1 - \sum_{i=1}^k p_{s_i1}\right) \log \frac{1 - \sum_{i=1}^k p_{s_i1}}{1 - \sum_{i=1}^k p_{s_i2}} = B_{12}^k \quad (2.30)$$

and

$$\begin{aligned} B_{21}^{k+1} & \geq \sum_{i=1}^k p_{s_i2} \log \frac{p_{s_i2}}{p_{s_i1}} + \left(1 - \sum_{i=1}^k p_{s_i2}\right) \log \frac{1 - \sum_{i=1}^k p_{s_i2}}{1 - \sum_{i=1}^k p_{s_i1}} \\ & = B_{21}^k \end{aligned} \quad (2.31)$$

Thus, combining Eq. (2.30) and Eq. (2.31), we have

$$\mathcal{J}_{s_{k+1}}^{k+1}(P_{s_{k+1}1}^{k+1}, P_{s_{k+1}2}^{k+1}) \geq \mathcal{J}_{s_k}^k(P_{s_k1}^k, P_{s_k2}^k) \quad (2.32)$$

□

This theorem indicates that the discriminative capacity increases when more features are used for classification, under the assumption that the term occurrence of a document satisfies a particular multinomial distribution. Note that the proposed greedy feature selection algorithm makes a locally optimal choice at each step to approximate the global optimal solution of Eq. (2.12), by selecting a feature with the maximum discriminative capacity for classification. The significance of this algorithm is that it starts at the best first feature and towards the optimal solution.

This greedy approach can be considered as a wrapper approach. However, unlike those existing wrapper approaches, this greedy approach does not need to evaluate the classification performance on a validation data set through retraining the classifier when a new feature is generated, because a closed form of KL-divergence is given in Eq. (2.19) to measure the discriminative capacity of MNB classifiers.

However, this greedy approach still has the computational complexity of $\mathcal{O}(M^2/2)$, which leads to a heavy computational load for a high-dimensional data set. Next, we provide a more efficient feature selection approach for text categorization.

2.3.3 An Efficient Feature Selection Approach

In Algorithm 1, the best one single feature is selected at the first step, providing an optimal starting point to approximate the optimal solution. At this step,

we rank the J-divergences over all features, which are given by

$$J_{e_1}^1(P_{e_1 1}^1, P_{e_1 2}^1) \geq J_{e_2}^1(P_{e_2 1}^1, P_{e_2 2}^1) \geq \cdots \geq J_{e_M}^1(P_{e_M 1}^1, P_{e_M 2}^1) \quad (2.33)$$

where e_i denotes the feature index and we know e_1 is s_1 which is given by Eq. (2.21) in Algorithm 1. Looking at the first two J-divergences in Eq. (2.33), we have

$$\begin{aligned} & p_{e_1 1} \log \frac{p_{e_1 1}}{p_{e_1 2}} + (1 - p_{e_1 1}) \log \frac{1 - p_{e_1 1}}{1 - p_{e_1 2}} \\ & \geq p_{e_2 1} \log \frac{p_{e_2 1}}{p_{e_2 2}} + (1 - p_{e_2 1}) \log \frac{1 - p_{e_2 1}}{1 - p_{e_2 2}} \end{aligned} \quad (2.34)$$

Since one single term probability is very small in practice, i.e., $p_{i1} \ll 1$ for $i = 1, 2, \dots, M$, the term $\log \frac{1-p_{i1}}{1-p_{i2}}$ would be very close to zeros, and then Eq. (2.34) may lead to

$$p_{e_1 1} \log \frac{p_{e_1 1}}{p_{e_1 2}} \geq p_{e_2 1} \log \frac{p_{e_2 1}}{p_{e_2 2}} \quad (2.35)$$

At the second step in Algorithm 1, the feature e_2 is usually selected due to the fact that $\log \frac{1-p_{e_1 1}-p_{e_2 1}}{1-p_{e_1 2}-p_{e_2 2}} \approx 0$. That is because, according to Eq. (2.35), we have

$$p_{e_1 1} \log \frac{p_{e_1 1}}{p_{e_1 2}} + p_{e_2 1} \log \frac{p_{e_2 1}}{p_{e_2 2}} \geq p_{e_1 1} \log \frac{p_{e_1 1}}{p_{e_1 2}} + p_{j 1} \log \frac{p_{j 1}}{p_{j 2}} \quad (2.36)$$

for $j = 1, 2, \dots, M, j \neq e_1, e_2$.

Therefore, instead of doing a greedy search, an efficient way is to use the ranked feature index set $\mathcal{E} = \{e_1, e_2, \dots, e_M\}$ in Eq. (2.33). We summarize this efficient feature selection algorithm in Algorithm 2. Compared to the Algorithm 1, the proposed approach is much more *efficient* as each feature score is calculated once, and it has the computational complexity of $\mathcal{O}(M)$. This efficient approach evaluates the “importance” of each individual feature by measuring its discriminative capacity, when only one single feature is used for classification. We note that the Theorem 1 is also satisfied for this efficient approach, i.e., the J-divergence

Algorithm 2: An Efficient Feature Selection Algorithm Based on the Maximum J-Divergence for Two-class Classification

INPUT:

- The estimated probabilities of M features: $\theta_i = [p_{1c}, p_{2c}, \dots, p_{Mc}]$, $c = 1, 2$ for two classes;

ALGORITHM:

for $i = 1 : M$ **do**

1. Form two variables: x_i and $\bar{x}_i$, and denote P_{i1} and P_{i2} for class 1 and 2 distribution, respectively;
2. Calculate J-divergence $J_i(P_{i1}, P_{i2})$ between P_{i1} and P_{i2} using Eq. (2.19) as score for the i -th feature;

end

3. Sort the feature scores in a descend way: $\mathcal{J}_{e_1} > \mathcal{J}_{e_2} > \dots > \mathcal{J}_{e_M}$.

OUTPUT:

- A ranked feature index set: $\mathcal{E} = \{e_1, e_2, \dots, e_M\}$.
-

measure increases as more features are selected. Meanwhile, we also note that the features selected in Algorithm 2, are not necessarily the same as ones selected in Algorithm 1, as one can see, for example, that Eq. (2.35) and Eq. (2.36) approximately hold for the first and second feature selection. Considering the first feature selection as an example, this approximation depends on the value of Δ defined as

$$\Delta = (1 - p_{e_21}) \log \frac{1 - p_{e_21}}{1 - p_{e_22}} - (1 - p_{e_11}) \log \frac{1 - p_{e_11}}{1 - p_{e_12}} \quad (2.37)$$

More precisely, given e_1 is selected as the first feature in Algorithm 2, e_1 also ranks first in Algorithm 1 if and only if the following condition holds:

$$\Delta \leq p_{e_11} \log \frac{p_{e_11}}{p_{e_12}} - p_{e_21} \log \frac{p_{e_21}}{p_{e_22}} \quad (2.38)$$

2.3.4 Multi-class Classification

In this section, we extend the above efficient feature selection method for multi-class classification problems. The efficient feature selection method for an N -class classification problem based on the maximum JMH-divergence is presented

Algorithm 3: An Efficient Feature Selection Algorithm Based on the Maximum JMH-Divergence for N -class Classification

INPUT:

- The estimated probabilities of M features: $\theta_c = [p_{1c}, p_{2c}, \dots, p_{Mc}]$, $c = 1, 2, \dots, N$;
- The prior probabilities of N classes: $p_1, p_2, \dots, p_N$;

ALGORITHM:

for $i = 1 : M$ **do**

1. Form two variables: x_i and $\bar{x}_i$, and denote P_{ic} as the class distribution, $c = 1, 2, \dots, N$;

for $c = 1 : N$ **do**

2. Form two distributions: P_{ic} and $\bar{P}_{ic}$, where $\bar{P}_{ic}$ is the one grouping all remaining $N - 1$ classes;

3. Calculate KL-divergence $\mathcal{KL}(P_{ic}, \bar{P}_{ic})$ between P_{ic} and $\bar{P}_{ic}$;

end

4. Calculate the JMH-divergence $\mathcal{JM}\mathcal{H}_i = \sum_{c=1}^N \mathcal{KL}(P_{ic}, \bar{P}_{ic})$ as the score for the i -th feature;

end

5. Sort the feature scores in a descend way:

$\mathcal{JM}\mathcal{H}_{e_1} > \mathcal{JM}\mathcal{H}_{e_2} > \dots > \mathcal{JM}\mathcal{H}_{e_M}$.

OUTPUT:

- A ranked feature index set: $\mathcal{E} = \{e_1, e_2, \dots, e_M\}$.
-

in Algorithm 3. The value of JMH-divergence is used as the score for each feature. Sorting the feature scores in a descend way, we output a ranked feature index set $\mathcal{E} = \{e_1, e_2, \dots, e_M\}$ for multi-class classification. The computational complexity of this algorithm is $\mathcal{O}(MN)$.

2.3.5 Feature Selection Based on χ^2 Statistics

The KL-divergence measure $\mathcal{KL}(P_1, P_2)$ is also known as the minimum discrimination information between the probability distributions P_1 and P_2 . Suppose that we have a random sample of l observations (e.g., a document with length l), and we try to test a null hypothesis H_2 , the observation is drawn from class 2 with

the distribution P_2 , against an alternative hypothesis H_1 , the observation is drawn from class 1 with the distribution P_1 . The minimum discrimination statistic [36] in favor of H_1 against H_2 is defined as

$$\mathcal{D}(\hat{P}_1, P_2) = \hat{P}_1 \log \frac{\hat{P}_1}{P_2} \quad (2.39)$$

where $\hat{P}_1$ is the estimate of P_1 from the given observations and P_2 is assumed to be known. We may reject the null hypothesis H_2 and accept the alternative hypothesis H_1 if the value of statistic $\mathcal{D}(\hat{P}_1, P_2)$ exceeds a predetermined threshold. Asymptotically, the statistic $2\mathcal{D}(\hat{P}_1, P_2)$ under the null hypothesis H_2 satisfies a central Chi-squared distribution χ_{M-1}^2 with $M-1$ degrees of freedom, and satisfies a non-central Chi-squared distribution $\chi_{M-1}^2(\nu)$ under the alternative hypothesis H_1 . The noncentrality parameter ν_D is given by

$$\nu_D(\hat{P}_1, P_2) = l \sum_{i=1}^M \frac{(\hat{p}_{i1} - p_{i2})^2}{p_{i2}} \quad (2.40)$$

where $\hat{p}_{i1}$ is the estimate of p_{i1} under H_1 using the MLE method from the given observations. Asymptotically, the J-divergence $\mathcal{J}(\hat{P}_1, P_2)$ is the sum of two χ^2 distributions. We have

$$\nu_J(\hat{P}_1, P_2) = \frac{l}{2} \sum_{i=1}^M \frac{(\hat{p}_{i1} - p_{i2})^2}{p_{i2}} + \frac{l}{2} \sum_{i=1}^M \frac{(\hat{p}_{i1} - p_{i2})^2}{\hat{p}_{i1}} \quad (2.41)$$

where the former one is known as the Pearson's χ^2 and the latter one is also known as the Neyman's χ^2 [36].

Thus, one can see that the noncentrality parameter ν_J would also be a good sign to indicate the discriminative capacity of discriminating between H_1 and H_2 . Therefore, we can further select the features by maximizing the noncentrality parameter ν_J for binary classification. Under the assumption that the number of samples in training data set goes infinity, we use the estimation of p_{i1} and p_{i2} for $i = 1, 2, \dots, M$ from training data in Eq. (2.41). For a multi-class

classification problem, unlike Algorithm 3, each feature is assigned with a score $\mathcal{CHI}_i = \sum_{c=1}^N \nu(P_{ic}, \bar{P}_{ic})$ which is the sum of N noncentrality parameters in their χ^2 distributions, and the ranked feature index set $\mathcal{S} = \{e_1, e_2, \dots, e_M\}$ is produced by sorting the scores:

$$\mathcal{CHI}_{e_1} > \mathcal{CHI}_{e_2} > \dots > \mathcal{CHI}_{e_M} \quad (2.42)$$

We need to note here that the feature selection approach based upon the Chi-squared statistic in Eq. (2.42) should be equivalent to the approach in Algorithm 3 if there are infinite training documents. When the assumption of large numbers is not satisfied, the features selected by Eq. (2.42) may lose some discriminative capacity. As we demonstrated through extensive experiments, the discrimination performance of the Chi-squared statistic is usually bounded by the approach in Algorithm 3. However, we sometimes also observe that the Chi-squared statistic performs better for some real-life text data sets. Therefore, in practice, we would like to recommend the use of both approaches because of their simplicity, efficiency and improved discrimination performance.

2.4 Experimental Results and Analysis

2.4.1 Real-Life Data Sets

For our experiments, we test our proposed feature selection approaches on two benchmarks that have been prepared by Deng et al. [47] [48] for text categorization: REUTERS and TOPIC DETECTION AND TRACKING (TDT2). These two benchmarks have been widely used in literature for performance evaluation.

The REUTERS originally contains 21,578 documents with 135 topics, but some documents belong to multiple topics. For our experiments, we use the ModApte version of the Reuters by removing those documents with multiple labels. This version consists of 8,293 documents in 65 topics. Following the work in [44], we form a data set, named REUTERS-20, consisting of the documents of the first 20

topics.

The TDT2 benchmark consists of 11,201 documents taken from two newswires (AP WorldStream and New York Times Newservice), two radio programs (PRI The World and VOA World News) and two television programs (CNN Headline News and ABC World News Tonight). Also, those documents that belong two or more topics have been removed. Because of the extremely imbalanced data for some categories, we only use the first 10 topics with the largest data size as our data set.

For these two data sets used in our experiments, we ignore those words in a stoplist, and discard those words that appear in less than 2 documents or messages in our preprocessing stage. For the data set of REUTERS-20, we perform classification on the officially split training and test data sets. For the data set of TDT2, we use 10-fold cross validation for performance evaluation, and the reported results are averaged over 10 runs.

2.4.2 Results

We compare our two efficient feature selection approaches: the maximum discrimination termed MD and its asymptotic χ^2 statistic termed $MD-\chi^2$, with the state-of-the-art feature ranking methods, including document frequency (DF), expected cross entropy for text (CET), χ^2 statistic and GSS . We carry out experiments on these two benchmarks when naive Bayes is used as classifiers. To compare the performance of these feature selection methods, we evaluate the classification accuracy of these classifiers with different number of features ranging from 10 to 2,000.

Fig. 2.1 shows the comparison results on two data sets of REUTERS-20 and TDT2. For the data set of REUTERS-20, it can be shown that our proposed two approaches with the first 200 selected features can achieve the similar performance

Figure 2.1: Accuracy on the data sets of (a). REUTERS-20 and (b). TDT2, when naive Bayes is used as the classifier.

as other four existing approaches with the first 1000 selected features. Moreover, the performance improvement of the proposed two approaches is increased in comparison with other methods. For the data set of TDT2, the proposed *MD* and *MD- χ^2* outperform all the others. It is interesting to notice that, when the first 100 features are selected, the *MD* obtains the accuracy of 96.46%, and the *MD- χ^2* has 95.76%. For other methods, such as *GSS* and χ^2 , the first 1000 features need to be selected to achieve the same classification accuracy.

2.5 Summary

Feature selection is an important feature representation method for the followed learning algorithms. In this chapter, we contribute a novel feature selection framework which aims to select the optimal feature subsets with maximum discrimination capability for classification. We show that feature's discrimination capability can be measured by the Jeffrey divergence asymptotically for two-class classification. For multi-class classification problems, we extend the Jeffrey divergence to a general one, named JMH-divergence, using one-vs-all scheme. Under this theoretical framework, several efficient feature selection algorithms are devel-

oped. Experimental results for text categorization have demonstrated that the features selected by the proposed feature selection methods can achieve superior prediction performance.

CHAPTER 3

EEF-Based Generative Learning

This chapter starts by giving an introduction of EEF in §3.1, a new PDF estimation method with the incorporation of known reference distributions. Several nice properties of EEF are given in §3.2, and a general classification rule is derived in §3.3 using the constructed distributions. Two types of classifiers are built in an either data-driven or model-driven manner in §3.4 and §3.5, respectively. The superior performance is further verified in several real-life data sets collected from different physical systems, including healthcare systems, bioinformatics, energy systems, and industrial infrastructures. With feature representation, information loss may always happen. To address this issue, §3.6 describes a generalized PDF projection theorem (GPPT) based on EEF and builds a classifier with class-specific features for classification.

3.1 Introduction to EEF

A data sample can be considered as an observation of a particular object of interest associated with a p -dimensional random variable $\mathbf{x}$. In the formalism of Gaussian processes [49][50] and probabilistic principal component analysis [51], it is assumed that training space points can be drawn from populations determined by the latent variable $\boldsymbol{\alpha}$, i.e., the class-wise distribution has the form $p(\mathbf{x}; \boldsymbol{\alpha})$. Following the definition of the generalized exponential family, as described below, it further assumes that the class-wise distribution $p(\mathbf{x}; \boldsymbol{\alpha})$ is an exponential family density in which $\boldsymbol{\alpha}$ are the natural parameters. For a member of the exponential family, the distribution takes the following form

$$p(\mathbf{x}; \boldsymbol{\alpha}) = \exp [\boldsymbol{\alpha}^T \mathbf{x} - G(\boldsymbol{\alpha})] p(\mathbf{x}; \mathbf{0}) \quad (3.1)$$

where the function $G(\cdot)$ is the so-called cumulant generating function which is defined as

$$G(\boldsymbol{\alpha}) = \ln \int_{\mathcal{X}} \exp(\boldsymbol{\alpha}^T \mathbf{x}) p(\mathbf{x}; \mathbf{0}) d\mathbf{x} \quad (3.2)$$

where $\mathcal{X}$ is the space of data vector $\mathbf{x}$. The cumulant generating function $G(\boldsymbol{\alpha})$ normalizes the $p(\mathbf{x}; \boldsymbol{\alpha})$ to be a probability density function. Under this assumption, we can move the data space $\mathcal{X}$ to the natural parameter space via the nonlinear transformation of cumulative generating function $G(\boldsymbol{\alpha})$.

In order to embed the data space of all classes into one natural parameter space, the embedded exponential family constructs the class-wise distribution using a reference distribution. Assume the reference distribution is denoted by $p_0(\mathbf{x})$, the constructed data distribution $p(\mathbf{x}; \boldsymbol{\eta})$ with natural parameters $\boldsymbol{\eta}$ is written as

$$p(\mathbf{x}; \boldsymbol{\eta}) = \exp [\boldsymbol{\eta}^T \mathbf{x} - K_0(\boldsymbol{\eta}) + \ln p_0(\mathbf{x})] \quad (3.3)$$

where $K_0(\boldsymbol{\eta})$ is also the cumulant generating function, defined as

$$K_0(\boldsymbol{\eta}) = \ln E_0 [\exp(\boldsymbol{\eta}^T \mathbf{x})] = \ln \int_{\mathcal{X}} \exp(\boldsymbol{\eta}^T \mathbf{x}) p_0(\mathbf{x}) d\mathbf{x} \quad (3.4)$$

which is the log-expectation of $\exp(\boldsymbol{\eta}^T \mathbf{x})$ with respect to the reference distribution. To avoid the confusion of natural parameters between exponential family and embedded exponential family, we will use $\boldsymbol{\eta}$ for the natural parameters of EEF in this chapter. The natural parameters $\boldsymbol{\eta}$, also named latent variables, determine the class-wise distribution.

When the PDF of the measurements $\mathbf{T} = \mathbf{t}(\mathbf{x})$ under a reference hypothesis $p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_0)$ is known, the EEF aims to construct a PDF $\hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta})$ which unifies the unknown PDFs under the reference hypothesis $\mathcal{H}_0$ and the M candidate hypotheses $\mathcal{H}_i, i = 1, \dots, M$. The PDF $\hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta})$ is indexed with the natural parameter vector $\boldsymbol{\eta}$. The measurements $\mathbf{T} = \mathbf{t}(\mathbf{x})$ are the sufficient statistics of $\mathbf{x}$. As stated in the

previous section, [52] demonstrates that the PDF $\hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta}_i)$ is the asymptotically closest to the true PDF $p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)$ under M hypotheses in the measurement of KL divergence. The KL divergence $D(p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i) || \hat{p}_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i))$ has been proved to determine the asymptotic performance for detection by Stein's Lemma [37] and for classification [53]. The Kullback-Leibler divergence [36] $D(p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i) || \hat{p}_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i))$ is written as

$$D(p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i) || \hat{p}_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)) = \int p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i) \ln \frac{p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)}{\hat{p}_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)} d\mathbf{t} \quad (3.5)$$

It is always non-negative and equals zero if and only if $p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i) = \hat{p}_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)$ for all $\mathbf{t}$. Thus, under the assumption of small signal, the PDF $\hat{p}_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)$ which approximates the true unknown PDF $p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)$ is found by minimizing the KL divergence in Eq. (3.5) with the constraint of moment matching [52]. The Kullback theorem [36] gives the solution for this problem, where $\hat{p}_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)$ can be written as

$$\hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta}) = \exp [< \boldsymbol{\eta}, \mathbf{t} > - K_0(\boldsymbol{\eta}) + \ln p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_0)] \quad (3.6)$$

where $\boldsymbol{\eta}$ is a $p \times 1$ vector for multivariate classification and $\boldsymbol{\eta} = [\eta_1, \eta_2, \dots, \eta_p]^T$, p denotes the dimension of the sample space, and $< \boldsymbol{\eta}, \mathbf{t} >$ is the inner product of $\boldsymbol{\eta}$ and $\mathbf{t}$ or $< \boldsymbol{\eta}, \mathbf{t} > = \boldsymbol{\eta}^T \mathbf{t}$. $K_0(\boldsymbol{\eta})$ is log-normalizer which makes the PDF integrate to one, written as

$$K_0(\boldsymbol{\eta}) = \ln E_0 [\exp(< \boldsymbol{\eta}, \mathbf{t} >)] = \ln \int_{\mathbf{t}} \exp(< \boldsymbol{\eta}, \mathbf{t} >) p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_0) d\mathbf{t} \quad (3.7)$$

Note that the PDF $\hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta})$ is indexed by the natural parameter vector $\boldsymbol{\eta}$ in Eq. (3.6). The natural parameter vector $\boldsymbol{\eta}$ differs from the source parameter vector $\boldsymbol{\theta}$ in the original PDF $p_{\mathbf{T}}(\mathbf{t}) = p_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\theta})$. Eq. (3.6) indicates that the reduced PDF no longer relates to the source parameters $\boldsymbol{\theta}$, but the natural parameters $\boldsymbol{\eta}$. Also, $\boldsymbol{\eta}$ is identifiable for $\hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta})$ since if $\boldsymbol{\eta}_1 \neq \boldsymbol{\eta}_2 \Rightarrow \hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta}_1) \neq \hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta}_2)$. Choosing $\boldsymbol{\eta} = \mathbf{0}$, the PDF is reduced to the reference hypothesis $\mathcal{H}_0$ PDF. Thus, all the

PDFs under reference hypothesis and M candidate hypotheses can be embedded into the reduced PDF with the natural parameter vector.

3.2 Properties of EEF

The PDF in the EEF inherits many useful algebraic properties of statistical exponential families [54]. Firstly, the moment generating function of EEF can be written as

$$m(s) = \exp [K_0(\boldsymbol{\eta} + s) - K_0(\boldsymbol{\eta})] \quad (3.8)$$

Secondly, the mean and variance matrix relates to the natural parameter vector $\boldsymbol{\eta}$ as

$$E(\mathbf{T}) = \nabla K_0(\boldsymbol{\eta}) \quad (3.9)$$

$$\text{Var}(\mathbf{T}) = \nabla^2 K_0(\boldsymbol{\eta}) \quad (3.10)$$

where $\nabla^2 K_0(\boldsymbol{\eta})$ denotes the Hessian of $K_0(\boldsymbol{\eta})$ and is a positive definite matrix since $K_0(\boldsymbol{\eta})$ is a strictly convex and differentiable function [55]. Hence, the $\langle \boldsymbol{\eta}, \mathbf{t} \rangle - K_0(\boldsymbol{\eta})$ in Eq. (3.6) is concave and the maximum of $\langle \boldsymbol{\eta}, \mathbf{t} \rangle - K_0(\boldsymbol{\eta})$ always exists. The maximum value is found either at the boundary points of the natural parameters, or at the point obtained by setting the differential to be zero

$$\frac{\partial K_0(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} = \mathbf{t} \quad (3.11)$$

3.3 EEF for Classification

In this section, we present how to build our classification rule based on the EEF PDF when the labeled data (training data) are available. For the PDF $\hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta})$ in Eq. (3.6), we define $l(\mathbf{t})$ as the log-likelihood function

$$l(\mathbf{t}; \boldsymbol{\eta}) = \ln \hat{p}_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta}) \quad (3.12)$$

Given a set of training data, we can estimate the natural parameter vector $\hat{\boldsymbol{\eta}}$ using the maximum likelihood estimator (MLE) [9]

$$\hat{\boldsymbol{\eta}} = \arg \max_{\boldsymbol{\eta}} l(\mathbf{t}; \boldsymbol{\eta}) \quad (3.13)$$

The estimated value $\hat{\boldsymbol{\eta}}_i$ for each hypothesis $\mathcal{H}_i, i = 1, 2, \dots, M$ can be obtained according to the Eq. (3.11). Then, we can write the PDF of measurements under $\mathcal{H}_i$ as

$$\hat{p}_{\mathbf{T}}(\mathbf{t}; \hat{\boldsymbol{\eta}}_i) = \exp[< \hat{\boldsymbol{\eta}}_i, \mathbf{t} > - K_0(\hat{\boldsymbol{\eta}}_i) + \ln p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_0)] \quad (3.14)$$

Since the reduced PDF is parametrized by the natural parameter vector $\boldsymbol{\eta}$ in EEF, we consider the $\hat{p}_{\mathbf{T}}(\mathbf{t}; \hat{\boldsymbol{\eta}}_i)$ equal to the estimation of $p_{\mathbf{T}}(\mathbf{t}; \mathcal{H}_i)$ under $\mathcal{H}_i$.

Unlike the discriminative approach, we employ the generative approach to maximize the posterior probability. Given unknown testing data $\mathbf{t}_s$ to be classified, similar to the MAP decision rule, we assign the class $\mathcal{H}_i$ to $\mathbf{t}_s$ if $p(\mathcal{H}_i|\mathbf{t}_s) \propto p(\mathbf{t}_s|\mathcal{H}_i)p(\mathcal{H}_i)$ is maximized over i . The prior probability for each class $p(\mathcal{H}_i)$ can be approximated by the ratio of number of training data for each class, and $p(\mathbf{t}_s|\mathcal{H}_i)$ can be replaced by its estimate $\hat{p}_{\mathbf{T}}(\mathbf{t}_s|\hat{\boldsymbol{\eta}}_i)$. Then, the target function of our classifier rule is built as following

$$\text{EEF}_i = < \hat{\boldsymbol{\eta}}_i, \mathbf{t}_s > - K_0(\hat{\boldsymbol{\eta}}_i) + \ln p(\mathcal{H}_i) \quad (3.15)$$

Thus, in the training process of our new classifier rule, the natural parameters of the EEF PDF are first estimated using the MLE from the sets of training data under each hypothesis. Then, the PDF for each hypothesis is available with the estimated natural parameter vectors $\hat{\boldsymbol{\eta}}$. In the testing process, the unknown testing data can be classified according to the target function built in Eq. (3.15), which is very similar to the MAP rule.

3.4 Data-Driven Classifier via EEF

3.4.1 Hypothesis

Given the sufficient statistics of data observations, the distribution in Eq. (3.14) can asymptotically approximate the true distribution for each class with respect to KL divergence, and the derived decision rule in Eq. (3.15) can provide the same classification performance as the MAP rule which usually provides an asymptotic upper bound of classification performance for all classifiers. However, for these general formulas in Eq. (3.14) and Eq. (3.15), it is hard to obtain the sufficient statistics without knowing the class-wise distributions. One way is that we use the data observation itself as one of its sufficient statistics, i.e. $\mathbf{T}(\mathbf{x}) = \mathbf{x}$, and thus, the reduced distribution can be rewritten as in Eq. (3.3).

As we mentioned above, the reference distribution $p_0(\mathbf{x})$ for the reference hypothesis $\mathcal{H}_0$ could be either known or unknown, and either Gaussian or non-Gaussian. Given the training data, we consider a more general case of non-Gaussian distribution for the reference hypothesis $\mathcal{H}_0$. We propose to use a Gaussian mixture model to represent the whole training data set. Assume the whole training data could be represented with a Gaussian mixture model with M components as follows

$$p_0(\mathbf{x}) = \sum_{m=1}^M \pi_m \mathcal{N}(\boldsymbol{\mu}_m, \mathbf{C}_m) \quad (3.16)$$

where π_m is the prior distribution for m -th Gaussian model, and $\mathcal{N}(\boldsymbol{\mu}_m, \mathbf{C}_m)$ represents the Gaussian distribution with mean vector $\boldsymbol{\mu}_m$ and covariance matrix $\mathbf{C}_m$. The expectation-maximization (EM) method is employed to estimate the parameters of the Gaussian mixture model with constraints that each Gaussian component only fits one class data under the reference hypothesis.

Unlike the classic classification method which uses the EM rule to estimate the Gaussian mixture model for each class, our classification rule constructs a new

distribution $p_{\mathbf{T}}(\mathbf{t}; \boldsymbol{\eta})$ with a reference distribution $p_0(\mathbf{x})$. The natural parameters $\boldsymbol{\eta}$ can be used to determine the class-wise distribution. In this data-driven manner, both the estimation of the reference distribution $p_0(\mathbf{x})$ and the class-wise natural parameters are estimated from the training data. While there are several Gaussian models underlying the training data set, the latent variable $\boldsymbol{\eta}$ in EEF can be considered as the variable indicating the probability that the observation $\mathbf{x}$ belongs to these models. Then, we use the training data from each class to estimate the class-wise natural parameter to build the final classification rule. Because the proposed method makes use of the information of the whole training data distribution, we could expect that the EEF PDF will perform better than Bayesian inference methods without using that information. To simplify our derivation, we also assume that the data variable is independent and identically distributed (i.i.d.), which means all observations are conditionally independent.

3.4.2 Distribution Construction via EEF

Given the training data, we assume that a Gaussian mixture model can represent the whole data distribution. Using the EM approach, the parameters of mean vector $\boldsymbol{\mu}_i$ and covariance matrix $\mathbf{C}_i$ for each class can be estimated. According to the definition of cumulant generating function in Eq. (3.4), we have

$$K_0(\boldsymbol{\eta}) = \ln E_0[\exp(\boldsymbol{\eta}^T \mathbf{x})] = \ln \sum_{m=1}^M \pi_m \int_{\mathcal{X}} \exp(\boldsymbol{\eta}^T \mathbf{x}) \mathcal{N}(\boldsymbol{\mu}_m, \mathbf{C}_m) d\mathbf{x} \quad (3.17)$$

For each Gaussian component in the above integral, we have

$$\begin{aligned}
\int_{\mathcal{X}} \exp(\boldsymbol{\eta}^T \mathbf{x}) \mathcal{N}(\boldsymbol{\mu}_m, \mathbf{C}_m) d\mathbf{x} &= \frac{1}{(2\pi)^{N/2} |\mathbf{C}_m|^{1/2}} \times \\
&\int_{\mathcal{X}} \exp \left[\boldsymbol{\eta}^T \mathbf{x} - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_m)^T \mathbf{C}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m) \right] d\mathbf{x} \\
&= \frac{1}{(2\pi)^{N/2} |\mathbf{C}_m|^{1/2}} \int_{\mathcal{X}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}'_m)^T \mathbf{C}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}'_m) \right] \times \\
&\quad \exp \left[\boldsymbol{\eta}^T \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\eta}^T \mathbf{C}_m \boldsymbol{\eta}^T \right] d\mathbf{x} \\
&= \exp \left[\boldsymbol{\eta}^T \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\eta}^T \mathbf{C}_m \boldsymbol{\eta}^T \right]
\end{aligned} \tag{3.18}$$

where $\boldsymbol{\mu}'_m = \mathbf{C}_m \boldsymbol{\theta} + \boldsymbol{\mu}_m$, and the property $\int p(\mathbf{x}) d(\mathbf{x}) = 1$ is used in the above derivation.

Replacing the above equation into Eq. (3.17), the final cumulant generating function $K_0(\boldsymbol{\eta})$ is obtained as

$$K_0(\boldsymbol{\eta}) = \ln \sum_{m=1}^M \pi_m \exp \left(\boldsymbol{\eta}^T \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\eta}^T \mathbf{C}_m \boldsymbol{\eta}^T \right) \tag{3.19}$$

Thus, the new distribution with EEF can be constructed as

$$\begin{aligned}
p(\mathbf{x}; \boldsymbol{\eta}) &= \exp \left[\boldsymbol{\eta}^T \mathbf{x} - K_0(\boldsymbol{\eta}) + \ln p_0(\mathbf{x}) \right] \\
&= \frac{\exp(\boldsymbol{\eta}^T \mathbf{x}) \sum_{m=1}^M \pi_m \mathcal{N}(\boldsymbol{\mu}_m, \mathbf{C}_m)}{\exp(K_0(\boldsymbol{\eta}))} \\
&= \frac{\sum_{m=1}^M \pi_m \exp \left[\boldsymbol{\eta}^T \mathbf{x} - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_m)^T \mathbf{C}_m^{-1} (\mathbf{x} - \boldsymbol{\mu}_m) \right]}{\sum_{m=1}^M \pi_m \exp(\boldsymbol{\eta}^T \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\eta}^T \mathbf{C}_m \boldsymbol{\eta}^T)}
\end{aligned} \tag{3.20}$$

Because the latent variable $\boldsymbol{\eta}$ determines the class-wise distribution, in the next step, we will estimate the natural parameter $\boldsymbol{\eta}$ with MLE given the training data for each class. Given N training data for one class, due to the assumption of i.i.d variable, we have

$$p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N; \boldsymbol{\eta}) = \prod_{i=1}^N p(\mathbf{x}_i; \boldsymbol{\eta}) = \exp \left[\boldsymbol{\eta}^T \sum_{i=1}^N \mathbf{x}_i - N K_0(\boldsymbol{\eta}) + \ln \prod_{i=1}^N p_0(\mathbf{x}) \right] \tag{3.21}$$

Similar to Eq. (3.11), the natural parameters $\boldsymbol{\eta}$ could be estimated by

$$\begin{aligned} \frac{\partial \ln p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N; \boldsymbol{\eta})}{\partial \boldsymbol{\eta}} &= 0 \\ \Rightarrow \frac{\partial K_0(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} &= \frac{\sum_{i=1}^N \mathbf{x}_i}{N} = \bar{\mathbf{x}} \end{aligned} \quad (3.22)$$

Given Eq. (3.19), we have

$$\frac{\partial K_0(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}} = \frac{\sum_{m=1}^M \pi_m (\boldsymbol{\mu}_m + \mathbf{C}_m \boldsymbol{\eta}) \exp(\boldsymbol{\eta}^T \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\eta}^T \mathbf{C}_m \boldsymbol{\eta})}{\sum_{m=1}^M \pi_m \exp(\boldsymbol{\eta}^T \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\eta}^T \mathbf{C}_m \boldsymbol{\eta})} \quad (3.23)$$

Combining Eq. (3.22) and Eq. (3.23), we further have

$$\sum_{m=1}^M \pi_m (\boldsymbol{\mu}_m + \mathbf{C}_m \boldsymbol{\eta} - \bar{\mathbf{x}}) \exp(\boldsymbol{\eta}^T \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\eta}^T \mathbf{C}_m \boldsymbol{\eta}) = 0 \quad (3.24)$$

Even though no closed form solution exists, a numerical solution can always be found by using an iterative approach because the $K_0(\boldsymbol{\eta})$ function is convex. Either the methods of searching for the zero point in Eq. (3.24) or of searching for the maximum point in Eq. (3.21) can be employed to find the parameters estimation $\hat{\boldsymbol{\eta}}$ for each class. For C -classes classification problem, $\hat{\boldsymbol{\eta}}_1, \hat{\boldsymbol{\eta}}_2, \dots, \hat{\boldsymbol{\eta}}_C$ can be estimated given the training data set.

Finally, given testing data $\mathbf{x}_t$ to be classified, we decide the class for which the following is maximized over all classes:

$$\text{EEF}_i = \hat{\boldsymbol{\eta}}_i^T \mathbf{x}_t - \ln \sum_{m=1}^M \pi_m \exp(\hat{\boldsymbol{\eta}}_i^T \boldsymbol{\mu}_m + \frac{1}{2} \hat{\boldsymbol{\eta}}_i^T \mathbf{C}_m \hat{\boldsymbol{\eta}}_i^T) + \ln p(\mathcal{H}_i) \quad (3.25)$$

3.4.3 Experiments

To demonstrate the generality of the proposed method, we test the classification performance for seven real-life data sets in the UCI repository [56], compared with the methods of pseudo-MAP, nearest neighbor and neural network. These data sets are summarized in Table 3.1 which shows the number of classes, the number of input dimensions, and the total number of data samples that are used in our experiments.

The data set *Knowledge* aims to predict students’ knowledge level about the subject of Electrical DC Machines [57]. The Single Proton Emission Computed Tomography (*SPECT*) is a medical image data set in which the patient is classified into two categories: normal and abnormal [58] [59]. With Indian Liver Patient Dataset (*ILPD*) the aim is to predict liver patient or not according to the age, gender, total proteins, etc. [60]. *Wine* attempts to determine the origin of wines using chemical analysis [61]. The data set *Abalone* is used to predict the age of abalone from physical measurements. Different ages are considered as 29 classes, but in our experiment, we treat it as 3-category classification problem by grouping classes 1 – 8 as category 1, classes 9 and 10 as category 2, and classes 11 and above as category 3. The *Seeds* data set aims to identify three different varieties of wheat: Kama, Rosa and Canadian, with 70 elements for each class. Finally, the data set *Vertebral* is used to classify orthopaedic patients into 2 classes (normal or abnormal) with 6 biomechanical features. The testing classification accuracy of the proposed method in comparison with pseudo-MAP rule, nearest neighbor and neural network is presented in Table 3.2, in which we highlight the best performance. It shows that the proposed EEF outperforms the other three methods for

Table 3.1: Description of seven UCI data sets

Dataset	# of classes	# of features	# of data
<i>Knowledge</i>	4	5	402
<i>SPECT</i>	2	22	266
<i>ILPD</i>	2	10	582
<i>Wine</i>	3	12	178
<i>Abalone</i>	3	8	4177
<i>Seeds</i>	3	7	210
<i>Vertebral</i>	2	6	310

Table 3.2: Average testing accuracy and standard derivation of the proposed method, Pseudo MAP, Nearest Neighbor, and Neural Network in percentage. The best value is highlighted with **Bold** value and the significant improvement is underlined for our proposed method.

Datasets	EEF	Pseudo MAP	Nearest Neighbor	Neural Network
<i>Knowledge</i>	82.73 ± 2.14	<u>81.69 ± 2.28</u>	<u>81.48 ± 2.12</u>	<u>81.04 ± 4.65</u>
<i>Spect</i>	81.46 ± 1.67	80.22 ± 2.13	<u>76.78 ± 2.36</u>	<u>78.40 ± 3.69</u>
<i>ILPD</i>	72.48 ± 2.86	<u>62.80 ± 2.22</u>	<u>63.48 ± 2.24</u>	<u>70.38 ± 2.21</u>
<i>Wine</i>	96.63 ± 1.09	96.28 ± 1.04	<u>94.88 ± 1.46</u>	<u>95.33 ± 1.89</u>
<i>Abalone</i>	66.25 ± 1.38	<u>63.98 ± 1.65</u>	<u>58.15 ± 1.07</u>	64.24 ± 1.40
<i>Seeds</i>	95.14 ± 1.67	<u>92.21 ± 3.07</u>	<u>89.22 ± 2.36</u>	<u>85.76 ± 4.87</u>
<i>Vertebral</i>	84.32 ± 2.45	<u>82.58 ± 3.09</u>	<u>81.77 ± 2.72</u>	82.89 ± 2.33

these seven real-life data sets. Table 3.2 shows the comparative results which are averaged over 100 runs. For each run, we randomly select half of data as training data, and the rest of data as test data. For the proposed EEF classifier rule used in our experiments, we use the AIC rule [62] to access the number of components M_i for each class and use $M = \sum_{i=1}^C M_i$ in our method. For the neural network, we employ a typical multi-layer perceptron (MLP) structure with the back propagation learning method. We use 10 neurons in the hidden layer, and set the learning rate of 0.01 with 1000 iterations. The interested reader is referred to [63] for further information. In addition, a bias neuron is used in both input layer and hidden layer. We also examine whether two results are significantly different according to a paired t -test with a significance level of 0.01.

3.5 Model-Driven Classifier via EEF: Power Quality Disturbance Classification

3.5.1 Hypothesis

In this section, we apply the EEF to power quality (PQ) disturbance pattern classification problem which is a long-standing critical issue in the electrical power

industry [64]. Recently, many methods have been proposed for improving the classification performance to decrease the economic loss, such as Self Organizing Learning Array (SOLAR) system based on wavelet transformation [65], inductive inference approach [66], SVM classification, etc. Power quality disturbance classification aims to recognize which type of disturbance occurred in a power system and take appropriate measurements. Here, we adopt the popular PQ disturbance models with seven different classes ($C1 - C7$) as shown in Table 3.3, where A denotes the amplitude of sine, ω_0 denotes the angular frequency of sine, and $u(t)$ denotes the step function. They include the normal, swell, sag, harmonic, outage, sag with harmonic, swell with harmonic [65][66] models.

Assuming that these power signals $s_i, i = 1, 2, \dots, 7$ are affected by additive Gaussian white noise, which is widely considered in the research of power quality fields [67] [68] [69], we write the observed power signal models as

$$x[n] = s[n] + w[n] \quad (3.26)$$

where w is assumed to be white Gaussian noise with PDF $w \sim \mathcal{N}(0, \sigma^2)$. Thus we can consider the following hypotheses for this power quality distribution classification problem

$$\begin{aligned} \mathcal{H}_0 : \quad & \mathbf{x} = \mathbf{w} \\ \mathcal{H}_i : \quad & \mathbf{x} = \mathbf{s}_i + \mathbf{w} \quad i = 1, 2, \dots, 7 \end{aligned} \quad (3.27)$$

where $\mathbf{x}$, $\mathbf{s}_i$ and $\mathbf{w}$ are all $N \times 1$ vectors, and $\mathcal{H}_0$ is considered as the reference hypothesis. Given one unknown disturbance, we aim to assign it to one of these hypotheses $\mathcal{H}_1, \dots, \mathcal{H}_7$.

Table 3.3: Categories of power quality disturbance.

PQ Disturbance Type	Class Symbol	Signal Model	Parameters
Normal	C_1	$s_1 = A \sin(\omega_0 t)$	N/A
Swell	C_2	$s_2 = A(1 + \alpha(u(t - t_1) - u(t - t_2))) \sin(\omega_0 t)$ $u(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$	$0.1 \leq \alpha \leq 0.8,$ $T \leq t_2 - t_1 \leq 9T$
Sag	C_3	$s_3 = A(1 - \alpha(u(t - t_1) - u(t - t_2))) \sin(\omega_0 t)$	$0.1 \leq \alpha \leq 0.8,$ $T \leq t_2 - t_1 \leq 9T$
Harmonic	C_4	$s_4 = A(\alpha_1 \sin(\omega_0 t) + \alpha_3 \sin(3\omega_0 t) +$ $\alpha_5 \sin(5\omega_0 t) + \alpha_7 \sin(7\omega_0 t))$	$0.05 \leq \alpha_3 \leq 0.15,$ $0.05 \leq \alpha_5 \leq 0.15,$ $0.05 \leq \alpha_7 \leq 0.15,$ $\sum \alpha_i^2 = 1$
Outrage	C_5	$s_5 = A(1 - \alpha(u(t - t_1) - u(t - t_2))) \sin(\omega_0 t)$	$0.9 \leq \alpha \leq 1,$ $T \leq t_2 - t_1 \leq 9T$
Sag with Harmonic	C_6	$s_6 = A(1 - \alpha(u(t - t_1) - u(t - t_2)))$ $(\alpha_1 \sin(\omega_0 t) + \alpha_3 \sin(3\omega_0 t) + \alpha_5 \sin(5\omega_0 t))$	$0.1 \leq \alpha \leq 0.9,$ $T \leq t_2 - t_1 \leq 9T,$ $0.05 \leq \alpha_3 \leq 0.15,$ $0.05 \leq \alpha_5 \leq 0.15,$ $\sum \alpha_i^2 = 1$
Swell with Harmonic	C_7	$s_7 = A(1 + \alpha(u(t - t_1) - u(t - t_2)))$ $(\alpha_1 \sin(\omega_0 t) + \alpha_3 \sin(3\omega_0 t) + \alpha_5 \sin(5\omega_0 t))$	$0.1 \leq \alpha \leq 0.9,$ $T \leq t_2 - t_1 \leq 9T,$ $0.05 \leq \alpha_3 \leq 0.15,$ $0.05 \leq \alpha_5 \leq 0.15,$ $\sum \alpha_i^2 = 1$

3.5.2 Reduced Distribution Construction

According to the power quality disturbance signal models shown in Table 3.3, we can also rewrite these signal models as follows

$$\mathbf{s}_i = \mathbf{H}_i \boldsymbol{\theta}_i \quad (3.28)$$

where $\boldsymbol{\theta}_i$ is a $p_i \times 1$ unknown vector by which the signal $\mathbf{s}_i$ under hypothesis $\mathcal{H}_i$ is parametrized, and $\mathbf{H}_i$ is a $N \times p_i$ known matrix for the i -th signal model. N is the number of data samples for all the signal models, and p_i is the number of unknown parameters for the i -th signal model since the number of unknown parameters under different signal model may not be equal.

By replacing the signal model in Eq. (3.28) by Eq. (3.27), we rewrite the signal models as

$$\begin{aligned} \mathcal{H}_0 : \quad \mathbf{x} &= \mathbf{w} \\ \mathcal{H}_i : \quad \mathbf{x} &= \mathbf{H}_i \boldsymbol{\theta}_i + \mathbf{w} \quad i = 1, 2, \dots, 7 \end{aligned} \quad (3.29)$$

which is a classification problem for the linear model. Note that because $\boldsymbol{\theta}_i$ represents the i -th hypothesis, we can randomly choose t_1 and t_2 in the step function $u(t_1) - u(t_2)$ for $\mathbf{H}_i, i = 2, 3, 5, 6, 7$ in Eq. (3.29). For this case, we choose the statistic $T(\mathbf{x}) = \mathbf{x}$. Then the PDF of $\mathbf{x}$ under the reference hypothesis $\mathcal{H}_0$ is

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}) \quad \text{under } \mathcal{H}_0 \quad (3.30)$$

and the log-normalized factor $K_0(\boldsymbol{\eta})$ is

$$K_0(\boldsymbol{\eta}) = \ln E_0[\exp(\boldsymbol{\eta}^T \mathbf{x})] = \frac{\sigma^2 \boldsymbol{\eta}^T \boldsymbol{\eta}}{2} \quad (3.31)$$

Thus, we construct the EEF PDF of $\mathbf{x}$ with EEF as

$$\begin{aligned} p(\mathbf{x}, \boldsymbol{\eta}) &= \exp [\boldsymbol{\eta}^T \mathbf{x} - K_0(\boldsymbol{\eta}) + \ln p_0(\mathbf{x})] \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp \left(-\frac{\mathbf{x}^T \mathbf{x}}{2\sigma^2} \right) \exp \left(\boldsymbol{\eta}^T \mathbf{x} - \frac{\sigma^2 \boldsymbol{\eta}^T \boldsymbol{\eta}}{2} \right) \end{aligned} \quad (3.32)$$

Note that the natural parameter vector $\boldsymbol{\eta}_i$ for each hypothesis $\mathcal{H}_i, i = 1, 2, \dots, 7$ can be estimated by maximizing $\boldsymbol{\eta}^T \mathbf{x} - K_0(\boldsymbol{\eta})$ over $\boldsymbol{\eta}$. There are also some implicit constraints in finding the MLE [52]. Since $\boldsymbol{\eta}_i$ is the only parameter which represents the signal under $\mathcal{H}_i$, we have

$$\boldsymbol{\eta}_i = \mathbf{H}_i \boldsymbol{\theta}_i \quad (3.33)$$

Under this constraint, the MLE of $\boldsymbol{\theta}_i$ and the estimation of $\boldsymbol{\eta}_i$ can be obtained.

We have

$$\hat{\boldsymbol{\theta}} = \frac{(\mathbf{H}_i^T \mathbf{H}_i)^{-1} \mathbf{H}_i^T \mathbf{x}}{\sigma^2} \quad (3.34)$$

and therefore

$$\hat{\boldsymbol{\eta}}_i = \frac{\mathbf{H}_i (\mathbf{H}_i^T \mathbf{H}_i)^{-1} \mathbf{H}_i^T \mathbf{x}}{\sigma^2} \quad (3.35)$$

Thus, the class-wise distribution can be rewritten as

$$p(\mathbf{x}, \hat{\boldsymbol{\eta}}) = \exp [\hat{\boldsymbol{\eta}}^T \mathbf{x} - K_0(\hat{\boldsymbol{\eta}}) + \ln p_0(\mathbf{x})] \quad (3.36)$$

3.5.3 Penalty Version of Classification Rule

Given one unknown power quality signal $\mathbf{x}$ to be classified, we decide $\mathcal{H}_i$ for which the following is maximized over i

$$\begin{aligned} \text{EEF}_i &= \hat{\boldsymbol{\eta}}_i^T \mathbf{x} - K_0(\hat{\boldsymbol{\eta}}_i) + \ln p(\mathcal{H}_i) \\ &= \hat{\boldsymbol{\eta}}_i^T \mathbf{x} - \frac{\sigma^2 \hat{\boldsymbol{\eta}}_i^T \hat{\boldsymbol{\eta}}_i}{2} + \ln p(\mathcal{H}_i) \\ &= \frac{\mathbf{x}^T \mathbf{H}_i (\mathbf{H}_i^T \mathbf{H}_i)^{-1} \mathbf{H}_i^T \mathbf{x}}{2\sigma^2} + \ln p(\mathcal{H}_i) \\ &= \frac{Q_i}{2} + \ln p(\mathcal{H}_i) \end{aligned} \quad (3.37)$$

where $Q_i = \mathbf{x}^T \mathbf{H}_i (\mathbf{H}_i^T \mathbf{H}_i)^{-1} \mathbf{H}_i^T \mathbf{x} / \sigma^2$. Note that the above constructed test statistic has the form of the GLRT for linear model [70]. It also can be realized as a ratio of generalized energy classifier since the orthogonal projection matrix $\mathbf{P}_H = \mathbf{H}_i (\mathbf{H}_i^T \mathbf{H}_i)^{-1} \mathbf{H}_i^T$ projects the data vector onto the signal sub-space [71].

We also notice that there are different numbers of unknown parameters for different classes. Because the value of Q_i monotonically decreases with respect to the number of unknown parameters, there is poorer classification performance for the class data with more unknown parameters. Instead of using Eq. (3.37) directly for PQ classification, we introduce a penalty version of classification rule which is similar to the EEF method for model selection [11]

$$\text{EEF}'_i = \begin{cases} \frac{Q_i}{2} - \frac{p_i}{2} \left(\ln \frac{Q_i}{p_i} + 1 \right) + \ln p(\mathcal{H}_i) & \text{if } Q_i > p_i \\ 0 & \text{otherwise} \end{cases} \quad (3.38)$$

We can also write this rule in a compact form

$$\text{EEF}'_i = \left(\frac{Q_i}{2} - \frac{p_i}{2} \left(\ln \frac{Q_i}{p_i} + 1 \right) + \ln p(\mathcal{H}_i) \right) u \left(\frac{Q_i}{p_i} - 1 \right) \quad (3.39)$$

where $u(x)$ is the unit step function. Since Q_i increases with the number of unknown parameters, the penalty term $p_i(\ln Q_i/p_i + 1)$ also increases with the number of unknown parameters.

3.5.4 Simulations

To evaluate the classification performance of our rule, we compare our method to three other existing methods: SOLAR based on wavelet feature extraction [65], inductive inference approach [66] and the SVM method. There are two kinds of SVM (C-SVM and ν -SVM) with many different kernel functions. Here we choose C-SVM with a sigmoid kernel function which has the best performance among these different SVM methods in PQ disturbance classification as demonstrated in [65]. In this simulation for PQ disturbance classification problem, let the angular frequency $\omega = 100\pi$ rad/s, the sampling frequency $F_s = 256$ Hz and the amplitude $A = 110$. We also use 200 training PQ disturbance signals to construct the PDF in Eq. (3.32) and estimate the natural parameter vectors in Eq. (3.34), and 200 testing PQ disturbance signals to evaluate the classification performance according to the target function in Eq. (3.37) for each hypothesis. The overall accuracy for

this seven class PQ disturbance problem is given in Table 3.4. In Table. 3.4, each 7×7 matrix shows the classification performance for each method, and the diagonal elements of each matrix denotes the number of correctly classified PQ disturbance signals among 200 PQ disturbance signals. It is easily shown that our rule has much better classification performance 98.29%, compared to 94.93% in SOLAR based on wavelet feature extraction, 90.4% in inductive inference approach, and 94.86% in C-SVM with sigmoid kernel function.

Table 3.4: Performance of power quality disturbance using EEF rule compared with the SOLAR, Inductive Inference and C-SVM.

EEF Rule with Penalty								SOLAR Based on Wavelet Feature Extraction							
	C_1	C_2	C_3	C_4	C_5	C_6	C_7		C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_1	200	0	0	0	0	0	0	C_1	200	0	0	0	0	0	0
C_2	1	199	0	0	0	0	0	C_2	0	200	0	0	0	0	0
C_3	0	0	190	4	6	0	0	C_3	1	0	174	0	24	1	0
C_4	0	0	0	200	0	0	0	C_4	0	0	0	200	0	0	0
C_5	0	0	5	3	192	0	0	C_5	15	0	16	0	161	8	0
C_6	0	0	0	4	0	196	0	C_6	0	0	2	1	2	194	1
C_7	0	0	0	1	0	0	199	C_7	0	0	0	0	0	0	200
Error	1.71%							Error	5.07%						
Inductive Inference Approach								C-SVM with Sigmoid Kernel Function							
	C_1	C_2	C_3	C_4	C_5	C_6	C_7		C_1	C_2	C_3	C_4	C_5	C_6	C_7
C_1	200	0	0	0	0	0	0	C_1	200	0	0	0	0	0	0
C_2	0	194	0	0	0	0	6	C_2	0	200	0	0	0	0	0
C_3	0	0	153	0	11	36	0	C_3	0	0	171	0	28	1	0
C_4	0	0	0	200	0	0	0	C_4	0	0	0	200	0	0	0
C_5	0	0	1	0	180	19	0	C_5	0	0	33	0	164	3	0
C_6	0	0	42	0	15	143	0	C_6	0	0	1	0	6	193	0
C_7	0	4	0	0	0	0	196	C_7	0	0	0	0	0	0	200
Error	9.60%							Error	5.14%						

3.6 GPPT: Generalized PDF Projection Theorem for Classification based on EEF

3.6.1 Motivations

As described in Chapter 2, it is necessary to perform feature reduction for high-dimensional data sets. By doing so, the PDFs can be estimated robustly in the low-dimensional feature subspace. However, feature reduction may lose pertinent information for discrimination. For example, data samples from different classes that could be well separated in the raw data space may be overlapped in the feature

subspace, causing classification errors.

The PDF reconstruction approach provides a solution to address this information loss issue in feature reduction by reconstructing the PDF on raw data and making classification in raw data space, which could improve classification performance. Several approaches have been developed along this track. Moghaddam et al. [72][73] use an eigenspace decomposition to approximate the high-dimensional raw data PDF, where the raw data space is divided into two complementary subspaces using Principal Components Analysis (PCA): the principal subspace (distance in feature space) and the orthogonal complement subspace (distance from feature space). While the PDF in the low-dimensional principal subspace is estimated using training data, the PDF in the complementary subspace is approximated with the PCA residual error. Then, the estimated PDF in the raw data space is written as the product of these two PDFs. More recently, researchers apply Bayesian partitioning techniques to estimate the distribution in high dimensional data space. For example, Wong and Ma in [74] developed the Optional Polya Tree (OPT) to construct a prior distribution, and Lu et al. in [75] derived a closed form of posterior probability using Bayesian sequential partitioning.

PDF Projection Theorem (PPT) [76] offers another solution for distribution construction which projects the PDF in the feature subspace back to the raw data space. It can be shown that all PDFs that generate the given feature PDF can be constructed with the PPT by selecting a suitable reference hypothesis. The generality of PPT makes it a good one for classification to avoid the “curse of dimensionality” [76][77]. It also allows class-specific features, that is to say, each class could have its own feature transformation function. Class-specific features offer many advantages for multi-class classification. For example, class-specific features carry much more discriminative information from the original raw data,

because each class can select the most discriminative features against the other classes. This characteristic makes the PPT different from many other classifications methods which usually need to incorporate a one-vs-all classification scheme [42] to build hierarchical multiclassifiers [78][79] to use class-specific features.

EEF is related to PPT. Like PPT, EEF is based on the estimated feature PDF and a specified reference hypothesis. While PPT produces a raw data PDF that reproduces the given feature PDF exactly, EEF is a way of combining one or more PDFs constructed with PPT in a geometric mixture with the reference hypothesis. The raw data PDF constructed using EEF reproduces the moments of a log-likelihood ratio statistic. This statistic can be easily estimated in the feature space and is directly linked to class separability. Motivated by this, we reconstruct the PDFs using EEF, which can be shown that the PPT is a special case of our new approach called generalized PPT (GPPT). Using the constructed PDFs on raw data, we derive a Bayesian classifier with class-specific features, termed GPPT classifier, and apply it for text categorization as a case study.

3.6.2 GPPT: PDF Construction via EEF

The Bayesian learning requires the estimation of class-wise distribution $p(\mathbf{x}|c_i)$ from training data. For high-dimensional data, it is impractical to estimate $p(\mathbf{x}|c_i)$ accurately when the given training data is limited. For this case, one usually reduces the sample $\mathbf{x}$ via feature transformation: $\mathbf{z} = f(\mathbf{x})$, where $\mathbf{z} \in \mathcal{R}^K$ is called the feature of $\mathbf{x}$ and the dimension of $\mathbf{z}$ is far less than that of $\mathbf{x}$, i.e., $K \ll D$. By doing so, the estimation of $p(\mathbf{z}|c_i)$ in the feature subspace is simplified. In this subsection, we show that the raw data PDF $p(\mathbf{x}|c_i)$ can be constructed from the feature PDF $p(\mathbf{z}|c_i)$ using EEF. First, we define a smoothing reference hypothesis c_0 (e.g., the union of all classes is used as c_0 in our study case), which is non-committal with respect to the N classes. Next we define a log-likelihood ratio

statistic $t = T(\mathbf{x}) = \log p(f(\mathbf{x})|c_i)/p(f(\mathbf{x})|c_0) = \log p(\mathbf{z}|c_i)/p(\mathbf{z}|c_0)$, which is a measure of the discriminative power between the given class and the reference hypothesis.

Mathematically, using EEF in Eq. (3.6), we estimate the PDF $p(\mathbf{x}|c_i)$ for class c_i in raw data space as follows:

$$p(\mathbf{x}|c_i; \eta) = \exp \left(\eta \ln \frac{p(\mathbf{z}|c_i)}{p(\mathbf{z}|c_0)} - K_0(\eta) + \ln p(\mathbf{x}|c_0) \right) \quad (3.40)$$

where η is the embedding parameter, and $K_0(\eta)$ is the cumulant generating function, which is given by:

$$\begin{aligned} K_0(\eta) &= \ln \int_{\mathbf{x}} \exp \left(\eta \ln \frac{p(\mathbf{z}|c_i)}{p(\mathbf{z}|c_0)} \right) p(\mathbf{x}|c_0) d\mathbf{x} \\ &= \ln E_{p_0} \left[\exp \left(\eta \ln \frac{p(\mathbf{z}|c_i)}{p(\mathbf{z}|c_0)} \right) \right] \end{aligned} \quad (3.41)$$

where $E_{p_0}[\cdot]$ denotes the expectation with respect to the distribution $p_0 = p(\mathbf{x}|c_0)$. Note that for $\eta = 1$, we have $K_0(\eta) = 0$ and $p(\mathbf{x}|c_i) = p(\mathbf{x}|c_0)/p(\mathbf{z}|c_0)p(\mathbf{z}|c_i)$, which is the PPT [76], and we call this distribution construction method as the generalized PPT (GPPT).

In GPPT, it is better to choose the reference distribution that is smooth and non-committal with respect to the N classes. The reference hypothesis consisting of the union of all classes is good one, and can be considered the geometric center of PDFs of all classes [80]. The embedding parameter η specifies the constructed PDF that has minimum KL-divergence to the reference distribution with the constraint of moment-matching. For each class, the optimal embedding parameter η_i^* can be estimated using the MLE criterion, which is given by:

$$\eta_i^* = \arg \max_{\eta} \eta \ln \frac{p(\mathbf{z}|c_i)}{p(\mathbf{z}|c_0)} - K_0(\eta) \quad i = 1, 2, \dots, N \quad (3.42)$$

Since the cumulant generating function $K_0(\eta)$ is strictly convex and differentiable, the target function in Eq. (3.42) is concave and the optimal embedding parameter η_i^* can be easily found.

3.6.3 GPPT for Classification with Class-Specific Features

The PDF construction on raw data from the PDF on features allows class-specific features for classification. Let $f_i(\mathbf{x})$ be the feature transformation for class i , and thus we have class-specific features $\mathbf{z}_i = f_i(\mathbf{x})$ for $i = 1, 2, \dots, N$. Using Eq. (3.40), for each class, we can always construct the PDF $p(\mathbf{x}|c_i; \eta_i)$ in raw data space from the PDF in class-specific feature space $p(\mathbf{z}_i|c_i)$. Applying the MAP rule, we make classification decisions as follows:

$$c^* = \arg \max_{i \in \{1, 2, \dots, N\}} \eta_i \ln \frac{p(\mathbf{z}_i|c_i)}{p(\mathbf{z}_i|c_0)} - K_0(\eta_i) + \ln p(c_i) \quad (3.43)$$

We note here that by using a common reference distribution in the PDF construction, the classifier given by Eq. (3.43) can be constructed without actually measuring the raw data $\mathbf{x}$. Nevertheless, Eq. (3.43) is based on an implied raw data PDF. One could apply a different reference distribution to the PDF construction of each class, which would require measuring $\mathbf{x}$, but this is not explored in this thesis.

We apply the proposed GPPT classifier for text categorization in which the multinomial naive Bayes (MNB) is used as classifier. In Fig. 3.1, we illustrate the difference between our GPPT classifier and the conventional classifier for text categorization. Using the “bag-of-words”, a document is transformed to a real-valued vector through a dictionary that consists of all distinct words or phrases for a data set. In the real-value vector, the element denotes the occurrence of words in the document. Because of its high dimensionality, it is necessary to perform feature reduction to reduce the computational burden for training a classifier. As described in previous sections, feature selection is a commonly used method for feature reduction in text categorization. In conventional approaches, a feature importance measurement, such as information gain (IG) [81] or maximum discrimination (MD) [82], is first employed to calculate feature importance for each individual class, and

then a global function, such as sum or weighted average, is applied to rank features to select a common feature subset for all classes. In contrast, we rank features for each class and apply the class-specific features for classification.

Figure 3.1: Flow chart of the GPPT classifier with class-specific features for text categorization (right), compared with the conventional approach (left).

In MNB, the features (word occurrences) of each class satisfy a specific multinomial distribution. Let $\mathbf{x} \in \mathcal{R}^D$ be the raw feature transformed from the document, and then for each class c_i , $i = 1, 2, \dots, N$, we have a multinomial distribution $p(\mathbf{x}|c_i)$ with D parameters (cell probabilities): $[p_{i,1}, \dots, p_{i,D}]$. The likelihood of observing a document $\mathbf{x}$ in class c_i conditioned on its document length l ¹ is

¹The likelihood of observing a document $p(\mathbf{x}|c_i, l)$ is conditioned on the document length l . This is different than the conventional MNB classifier for text categorization in which the document length is usually assumed to be constant, i.e., $p(\mathbf{x}|c_i, l) = p(\mathbf{x}|c_i)$.

given by:

$$p(\mathbf{x}|c_i, l) = \frac{l!}{x_1!x_2!\cdots x_D!} \prod_{k=1}^{D-1} p_{i,k}^{x_k} p_{i,D}^{x_D} \quad (3.44)$$

where $\sum_{k=1}^D p_{i,k} = 1$ and $\sum_{k=1}^D x_k = l$.

Suppose that the feature selection will select K out of D features. Denote $\mathbf{z}_i$ as the feature vector in class c_i and $\mathbf{I}_i = [n_1^i, \dots, n_K^i]$ as the corresponding feature indexes in $\mathbf{x}$ such that $z_{ik} = x_{n_k^i}$. Note that the marginal distribution $p(\mathbf{z}_i|c_i)$ still satisfies a multinomial distribution, but with $K+1$ elements. The $(K+1)$ -st feature is the combination of all other features in $\mathbf{x}$ except for the K selected features, and the multinomial distribution $p(\mathbf{z}_i|c_i)$ has $K+1$ cells: $[p'_{i,1}, \dots, p'_{i,K}, p'_{i,K+1}]$ where $p'_{i,k} = p_{i,n_k^i}$, $k = 1, 2, \dots, K$, and $p'_{i,K+1} = 1 - \sum_{k=1}^K p'_{i,k}$.

We denote class c_0 as the reference class which consists of all given training data so that the reference distribution $p(\mathbf{x}|c_0)$ still satisfies a multinomial distribution with D parameters: $[p_{0,1}, \dots, p_{0,D}]$, each of which can be written as:

$$p_{0,k} = \sum_{i=1}^N p_{i,k} p(c_i) \quad k = 1, 2, \dots, D \quad (3.45)$$

Using the general construction form in Eq. (3.40), we construct the PDF $p(\mathbf{x}|c_i)$ for class c_i , $i = 1, 2, \dots, N$ as follows:

$$p(\mathbf{x}|c_i, l; \eta_i) = \exp \left[\eta_i \sum_{k=1}^K z_{ik} \beta_{ik} - K_1(\eta_i, l) + \ln p(\mathbf{x}|c_0) \right] \quad (3.46)$$

where

$$K_1(\eta_i, l) = l \ln \left(\sum_{k=1}^K p'_{0,k} \exp(\eta_i \beta_{ik}) + \left(1 - \sum_{k=1}^K p'_{0,k} \right) \right) \quad (3.47)$$

and

$$\beta_{ik} = \ln \frac{p'_{i,k}}{p'_{0,k}} - \ln \frac{p'_{i,K+1}}{p'_{0,K+1}} \quad (3.48)$$

Note that we obtain a closed form solution of the PDF construction in the original high-dimensional space of $\mathbf{x}$ as shown in Eq. (3.46) to Eq. (3.48).

Given a N -class training data set $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \dots \cup \mathcal{X}_N$, each class consists of M_i documents $\mathcal{X}_i = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{M_i}\}$, and each document $\mathbf{x}_m$ has a length of $l_m = \sum_{k=1}^D x_{mk}$, where x_{mk} is the k -th element in $\mathbf{x}_m$. We use the MLE to estimate the optimal embedding parameter, which is given by:

$$\eta_i^* = \arg \max_{\eta_i} \sum_{k=1}^K \bar{z}_{ik} \beta_{ik} - K_1(\eta_i, \bar{l}) \quad (3.49)$$

where $\bar{z}_{ik}$ and $\bar{l}$ are the average of word occurrences for the k -th selected feature and the average of the document length over the training set $\mathcal{X}_i$ of class c_i , respectively. Although it is difficult to find an analytic solution of η_i^* , it can be easily found using convex optimization techniques since the objective function is a concave function with respect to η_i .

3.6.4 Experimental Results and Analysis

We use the real-life data set of REUTERS-20 to evaluate the performance of our proposed GPPT approach for text categorization. In REUTERS-20, we have an original feature size of 18,933. To reduce the feature size, we apply the IG metric [81] to evaluate the feature importance. For each class c_i , the score of the k -th feature is calculated as follows:

$$IG(t_k, c_i) = p(t_k, c_i) \log \frac{p(t_k, c_i)}{p(t_k)p(c_i)} + p(\bar{t}_k, c_i) \log \frac{p(\bar{t}_k, c_i)}{p(\bar{t}_k)p(c_i)} \quad (3.50)$$

where t_k indicates the k -th term appears in the document, and $\bar{t}_k$ indicates it does not. It is shown that $IG(t_k, c_i)$ is a class-specific feature score. In conventional approaches, a global function, e.g., sum or average, is used to calculate class-independent feature scores for feature ranking, as shown in Fig. 3.1. However, the class-specific feature based classifiers rank the feature of each class with the score $IG(t_k, c_i)$ in Eq. (3.50), and use the class-specific features for classification.

We compare our GPPT class-specific MNB classifier with three other state-of-the-art classifiers: MNB classifier [43], support vector machine (SVM) [83][84], and PPT class-specific MNB classifier [76]. While the first two are commonly used in text categorization with class-independent features, the last one and our classifier use class-specific features for classification. In PPT class-specific MNB classifier, we use the same reference hypothesis given by Eq. (3.45) and class-specific features given by Eq. (3.50) as used in EEF, and make the classification decision with the following rule:

$$c^* = \arg \max_{i=\{1,2,\dots,N\}} \sum_{k=1}^{K+1} z_{ik} \ln \frac{p'_{i,k}}{p_{0,k}} + \ln p(c_i) \quad (3.51)$$

We report the classification results on the data set of REUTERS-20 in Fig. 3.2, where the feature size ranges from 100 to 2000. It can be shown that our GPPT class-specific MNB classifier outperforms the other three methods. For the REUTERS-20 data set, it can be shown that our GPPT class-specific MNB classifier has promising performance improvement with a large margin compared to the others.

Figure 3.2: Classification results on REUTERS-20.

3.7 Summary

Given a set of training data, a new generalized PDF with respect to the reference distribution can be constructed via EEF. The new distribution inherits many useful properties from the exponential family. It can be also used for classification in either a data-driven manner or a model-driven manner. The proposed methods could benefit many classification problems, and provide an effective way to address the problem of model parameters learning using embedded latent variable exponential family probability for classification. To address the information loss issue in feature reduction, a GPPT classifier based on EEF can be also built, which offers a novel way for classification with class-specific features. Several different real-life classification examples are considered in this chapter, and it can be shown that the proposed EEF classifier obtains competitive results compared with the state-of-the-art methods.

CHAPTER 4

ENN-Based Discriminative Learning

In many cyber-physical systems, vast amounts of data have high complexity and no single density distribution could model the data very well. The generative classifiers with the assumption of density distribution models may perform poorly if such assumption is inappropriate. This chapter describes several nearest neighbors based discriminative learning approaches. Particularly, in §4.1, ENN classification method and its several different versions are built for classification. In §4.2, a novel anomaly detection method based on the concept of extended nearest neighbors is proposed.

4.1 ENN: Extended Nearest Neighbor Method for Classification

4.1.1 Background and Motivations

Nearest neighbor-based pattern recognition methods have several key advantages, such as easy implementation, competitive performance, and a nonparametric computational basis which is independent of the underlying data distribution. The first modern study of the nearest neighbor approach can be traced back to 1951 by Fix and Hodges [21]. In their formalization of nonparametric discrimination, the consistency of KNN was established using a probability density estimation. They proved that if $k \rightarrow \infty$ and $k/n \rightarrow 0$, where k is the number of selected nearest neighbors and n is the sample size of the whole data set, the classification error of the nearest neighbor method (R_n) can asymptotically converge to the Bayesian rule (R^*): $R_n \rightarrow R^*$. In another representative work [22], Cover and Hart proved that for N -class classification problems, when $k = 1$, the classification error of the nearest neighbor method will be bounded by $R^* \leq R_n \leq R^*(2 - \frac{N}{N-1}R^*)$ when there are infinite samples. This basically means in the large sample case, the near-

est neighbor method has “a probability of error which is less than twice the Bayes probability of error” [22]. These early classic works laid down a strong theoretical foundation for nearest neighbor based methods, which have been witnessed in considerable applications in many different disciplines, such as biological and chemical data analysis [23], disease classification and clinical outcome prediction [24], among others.

In general, two types of errors may occur in KNN based methods: for the samples in the areas of higher density, the k nearest neighbors could lie in the areas of slightly lower density, or vice versa [26]. These types of errors may result in misclassification, especially when nearest neighbors are dominated by the samples from other classes. Error rates are increased when sample sizes are small or data are imbalanced (see a comprehensive survey on imbalanced learning [85]). Here, we introduce a new method, extended nearest neighbor (ENN), based on generalized class-wise statistics which can represent intra-class coherence. Unlike the classic KNN method where only the nearest neighbors are considered for classification, our proposed ENN method takes advantage of all available training data to make a classification decision: it assigns a class membership to an unknown sample to maximize the intra-class coherence. By exploiting the information from all available data to maximize the intra-class coherence, ENN is able to learn from the global distribution, therefore, improving pattern recognition performance.

4.1.2 Problem Formulation: Limitations of KNN

The reason of the “two types of errors” [26] is that KNN method is sensitive to the scale or variance of the distributions of the predefined classes [25]. The nearest neighbors of an unknown observation will tend to be dominated by the class with the highest density. For example, in a two-class classification problem, if we assume that class 1 has a smaller variance than class 2, then the class 1 samples dominate

their near neighborhood with higher density (i.e., more concentrated distribution), whereas the class 2 samples are distributed in regions with lower density (i.e., more spread out distribution). In this case, for class 2 samples, the number of nearest neighbors from class 1 may exceed that from class 2. For instance, for those class 2 samples which are close to the region of class 1, there may be a large number of class 1 neighbors because of the higher density of class 1. Only for those class 2 samples which are far away from the region of higher density of class 1, their nearest neighbors from class 2 will be dominant.

Fig. 4.1 shows one example (a two-class classification scenario) of decision making in the classic KNN method for two Gaussian distributions with different means and variances, in which we assume that their prior distribution $p(\omega)$ are the same. In this figure, the x -axis represents the data value (one dimensional data in this case), and the y -axis represents the class-conditional probability $p(x|\omega)$. For the eight data points illustrated here, we assume that data points x_1, x_2, x_3 and x_4 are sampled from the class 1 distribution, and data points x_5, x_6, x_7 and x_8 are sampled from the class 2 distribution. On the side of this figure, we also listed their corresponding k_1/k and k_2/k ratio (here $k = 5$ is the parameter of KNN, and k_1 and k_2 are the number of nearest neighbors belong to class 1 and class 2, respectively). In this case, the Bayesian method can correctly classify all these eight data points by calculating their corresponding posterior probabilities according to the Bayes' theorem. However, under the KNN method, all of these eight data points are predicted as class 1. This means that the four data points from class 2, x_5, x_6, x_7 and x_8 , are misclassified because their k_1/k ratio values are larger than their corresponding k_2/k ratio values. The reason is that the nearest neighbors of these four data points are dominated by the class 1 data because of the distribution. This has been a long-standing limitation of the classic KNN

Figure 4.1: One example of the KNN rule in comparison with the Bayesian rule for a two-class classification problem.

method in literature [26] [25].

4.1.3 ENN Classifier

Here we describe the Extended Nearest Neighbor (ENN) method to solve this deficiency in the classic KNN method. The advantages of the original KNN approach are retained in our ENN classifier, such as easy implementation and competitive classification performance. However, unlike the classic KNN approach, the new *ENN classifier makes a prediction by not only considering who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors.*

We describe our ENN method starting with the *two-class classification* problem. The generalized statistic based on nearest neighbors was first proposed in the two-sample test problem to evaluate whether two distributions are mixed well or widely spread [86], [87]. In our approach, unlike the generalized statistic over all samples as discussed in [87], we first build our generalized class-wise statistic regarding the pooled samples S_1 and S_2 for each class along with its nearest neigh-

bors. We define the generalized class-wise statistic T_i for class i as the following:

$$T_i = \frac{1}{n_i k} \sum_{\mathbf{x} \in S_i} \sum_{r=1}^k I_r(\mathbf{x}, S = S_1 \cup S_2) \quad i = 1, 2 \quad (4.1)$$

where S_1 and S_2 denote the samples in class 1 and class 2, respectively, $\mathbf{x}$ denotes one single sample in $S = S_1 \cup S_2$, n_i is the number of samples in S_i , and k is the user-defined parameter of the number of the nearest neighbors. The indicator function $I_r(\mathbf{x}, S)$ indicates whether both the sample $\mathbf{x}$ and its r -th nearest neighbor belong to the same class, defined as follows:

$$I_r(\mathbf{x}, S) = \begin{cases} 1, & \text{if } \mathbf{x} \in S_i \text{ and } \text{NN}_r(\mathbf{x}, S) \in S_i \\ 0, & \text{otherwise} \end{cases} \quad (4.2)$$

where $\text{NN}_r(\mathbf{x}, S)$ denotes the r -th nearest neighbor of $\mathbf{x}$ in S . This equation means for either class, if both the sample $\mathbf{x}$ and its r -th nearest neighbor in the pool of S belong to the same class, then the outcome of the indicator function $I_r(\mathbf{x}, S)$ equals 1; otherwise, it equals 0. In this way, the generalized class-wise statistic T_i in Eq. (4.1) is the ratio of the number of nearest neighbors belonging to the same class for class i with respect to the product of the sample size of class i (i.e., n_i) and the number of nearest neighbors under consideration (k). A large T_i indicates the samples in S_i are much closer together and their nearest neighbors are dominated by the same class samples, whereas a small T_i indicates that samples in S_i have an excess of nearest neighbors from the other class. Note that the generalized class-wise statistic has $0 \leq T_i \leq 1$ with $T_i = 1$ when all the nearest neighbors of class i data are also from the same class i , and $T_i = 0$ when all the nearest neighbors are from other classes. Based on this discussion, we can use T_i to represent the data distribution across multiple classes. Therefore, we introduce the concept of *intra-class coherence*, defined as follows:

$$\Theta = \sum_{i=1}^2 T_i \quad (4.3)$$

Given an unknown sample Z to be classified, we iteratively assign it to class 1 and class 2, respectively, to obtain two new generalized class-wise statistics T_i^j , where $j = 1, 2$.

$$T_i^j = \frac{1}{n_i' k} \sum_{\mathbf{x} \in S_{i,j}'} \sum_{r=1}^k I_r(\mathbf{x}, S' = S_1 \cup S_2 \cup \{Z\}) \quad i, j = 1, 2 \quad (4.4)$$

where n_i' is the size of $S_{i,j}'$, and $S_{i,j}'$ is defined as

$$S_{i,j}' = \begin{cases} S_i \cup \{Z\}, & \text{when } j = i \\ S_i, & \text{when } j \neq i \end{cases} \quad (4.5)$$

For this two-class classification problem, we have four generalized class-wise statistics: T_1^1 , T_2^1 , T_1^2 and T_2^2 . Unlike the previous KNN classifiers which predict that the unknown sample Z belongs to a class based only on its nearest neighbors, the ENN classifier predicts its class membership according to the following target function:

$$f_{\text{ENN}} = \arg \max_{j \in 1,2} \sum_{i=1}^2 T_i^j = \arg \max_{j \in 1,2} \Theta^j \quad (4.6)$$

where

$$\Theta^j = \sum_{i=1}^2 T_i^j \quad (4.7)$$

This means that our ENN method makes the prediction based on which decision resulting the largest intra-class coherence when we iteratively assume the test sample Z to be each possible class.

We now present the detailed calculation steps for a two-class classification problem using our ENN method. First, let's assume that the observation Z should be classified as class 1. Then, for each training sample, we re-calculate its k nearest neighbors and obtain the new generalized class-wise statistic according to Eq. (4.4), denoted as T_1^1 for class 1 and T_2^1 for class 2. Second, we assume that

the observation Z should be classified as class 2; then, for each training sample, we also re-calculate its k nearest neighbors and obtain the new generalized class-wise statistic according to Eq. (4.4), denoted as T_1^2 for class 1 and T_2^2 for class 2. Then a classification decision is made according to Eq. (4.6).

The aforementioned two-class ENN decision rule can be easily extended to *multi-class classification* problems. Specifically, for an N -class classification problem, we have the following algorithm:

ENN Classifier: Given an unknown sample Z to be classified, we iteratively assign it to each possible class j , $j = 1, 2, \dots, N$, and compute the generalized class-wise statistic T_i^j for each class i , $i = 1, 2, \dots, N$. Then, the sample Z is classified according to:

$$f_{\text{ENN}} = \arg \max_{j \in 1, 2, \dots, N} \sum_{i=1}^N T_i^j \quad (4.8)$$

4.1.4 ENN.V1 Classifier: An Equivalent Version of ENN

The underlying idea of the ENN rule shown in Eq. (4.8) is that we classify the test sample Z based on which decision results in the greatest intra-class coherence among all possible classes. We can further demonstrate that the classification of Z in our ENN rule depends not only on who are the nearest neighbors of Z , but also on who consider Z as one of their nearest neighbors.

For the new observation Z , let's assume that there are k_1 nearest neighbors from class 1 and k_2 nearest neighbors from class 2, where $k_1 + k_2 = k$ is the total number of nearest neighbors investigated. First, let's assume the observation Z to be classified as class 1; then, we count the number of class 1 data who have an *increased* number of class 1 samples in its k nearest neighbors (denote this number as Δn_1^1), and we also count the number of class 2 data who have a *decreased*

number of class 2 samples in its k nearest neighbors (denote this number as Δn_2^1). Then, in a similar way, we further assume the observation Z to be a member of class 2, and count the number of class 1 data who have a *decreased* number of class 1 samples in its k nearest neighbors (denote this number as Δn_1^2), and we also count the number of class 2 data who have an *increased* number of class 2 samples in its k nearest neighbors (denote this number as Δn_2^2). In this way, Δn_i^j represents how many of the k nearest neighbors from each class will change because of the introduction of the new sample Z , when it is iteratively assumed to be each possible class.

With these discussions, we now present an equivalent version of the ENN classifier as follows:

ENN.V1: Given an unknown sample Z to be classified, we iteratively assign it to each possible class j , $j = 1, 2, \dots, N$, and predict the class membership according to:

$$f_{\text{ENN.V1}} = \arg \max_{j \in 1, 2, \dots, N} \left\{ \left(\frac{\Delta n_i^j + k_i - kT_i}{(n_i + 1)k} \right)_{i=j} - \sum_{i \neq j}^N \frac{\Delta n_i^j}{n_i k} \right\} \quad (4.9)$$

where k is the user-defined parameter of the number of the nearest neighbors, n_i is the number of training data for class i , k_i is the number of the nearest neighbors of the test sample Z from class i , Δn_i^j represents the change of the k nearest neighbors for class i when the test sample Z is assumed to be class j , and T_i represents the generalized class-wise statistic of original class i (i.e., without the introduction of the test sample Z).

Proof of ENN.V1. We iteratively assign the unknown sample Z to class j to obtain new generalized class-wise statistic T_i^j for class i . According to Eq. (4.4), we have,

when $i = j$

$$\begin{aligned}
T_i^j &= \frac{1}{n'_i k} \sum_{X \in S_i \cup \{Z\}} \sum_{r=1}^k I_r(X, S' = S_1 \cup S_2 \cup \{Z\}) \\
&= \frac{1}{(n_i + 1)k} \left[\sum_{X \in S_i} \sum_{r=1}^k I_r(X, S_1 \cup S_2 \cup \{Z\}) + \sum_{r=1}^k I_r(Z, S_1 \cup S_2 \cup \{Z\}) \right] \\
&= \frac{1}{(n_i + 1)k} \left[\sum_{X \in S_i} \sum_{r=1}^k I_r(X, S_1 \cup S_2) + \Delta n_i^j + \sum_{r=1}^k I_r(Z, S_1 \cup S_2) \right] \\
&= \frac{1}{(n_i + 1)k} (n_i k T_i + \Delta n_i^j + k_i) \tag{4.10}
\end{aligned}$$

and when $i \neq j$

$$\begin{aligned}
T_i^j &= \frac{1}{n'_i k} \sum_{X \in S_i} \sum_{r=1}^k I_r(X, S' = S_1 \cup S_2 \cup \{Z\}) \\
&= \frac{1}{n_i k} \left[\sum_{X \in S_i} \sum_{r=1}^k I_r(X, S_1 \cup S_2) - \Delta n_i^j \right] \\
&= \frac{1}{n_i k} (n_i k T_i - \Delta n_i^j) \\
&= T_i - \frac{\Delta n_i^j}{n_i k} \tag{4.11}
\end{aligned}$$

Therefore, from Eq. (4.8), we have:

$$\begin{aligned}
f_{\text{ENN}} &= \arg \max_{j \in 1, 2, \dots, N} \sum_{i=1}^N T_i^j \\
&= \arg \max_{j \in 1, 2, \dots, N} \sum_{i=1}^N (T_i^j - T_i) \\
&= \arg \max_{j \in 1, 2, \dots, N} \left\{ (T_i^j - T_i)_{i=j} + \sum_{i \neq j}^N (T_i^j - T_i) \right\} \\
&= \arg \max_{j \in 1, 2, \dots, N} \left\{ \left(\frac{\Delta n_i^j + k_i - k T_i}{(n_i + 1)k} \right)_{i=j} - \sum_{i \neq j}^N \frac{\Delta n_i^j}{n_i k} \right\} \\
&= f_{\text{ENN.V1}} \tag{4.12}
\end{aligned}$$

□

Figure 4.2: (A), (B) and (C) demonstrate the detailed procedure of two versions of ENN classifier. In (A), training samples are stored and their k nearest neighbors and distances are calculated to compute the generalized class-wise statistic T_1 and T_2 . (B) and (C) illustrate the changes of the generalized class-wise statistic for each class, when we iteratively assume Z to be class 1 and class 2. Both of them provide the same classification result, but the recalculation of generalized class-wise statistic T_i^j is avoided in ENN.V1.

This proves our ENN.V1 is equivalent to the original ENN method. We now present an example in Fig. 4.2 for the detailed implementation of our two ENN versions, i.e., f_{ENN} in Eq. (4.8) and $f_{\text{ENN.V1}}$ in Eq. (4.9). Note that this equivalent target function $f_{\text{ENN.V1}}$ can provide the same classification result, without the recalculation of T_i^j as in f_{ENN} . Both target functions provide a general formula for

our ENN method for multi-class classification problems. In practical applications, whether using f_{ENN} or $f_{\text{ENN.V1}}$ is a user's preference. We would like to note that f_{ENN} is more straightforward to calculate and implement, but from a computational point of view, we recommend using $f_{\text{ENN.V1}}$ as this target function does not require recalculating the generalized class-wise statistic T_i^j for every test sample.

4.1.5 ENN.V2 Classifier: An Approximate Version of ENN

The proposed ENN classifier can resolve the scale-sensitive problem in the classic KNN decision rule as we discussed previously. From Eq. (4.8), the ENN method classifies the test sample Z based on which decision results in the greatest intra-class coherence among all predefined classes. Moreover, from Eq. (4.9), the classification decision rule of our ENN method is based on both the samples who are the nearest neighbors of Z and who regard Z as one of their nearest neighbors. Here, we provide an approximate but straightforward version of the ENN method under certain assumptions, named ENN.V2.

ENN.V2: Given an unknown sample Z to be classified, under the following two conditions:

- (1) All classes have the same number of data samples n , i.e., a balanced classification problem;
- (2) For all $i \neq j$, $\frac{\Delta n_i^j}{(n+1)nk} \rightarrow 0$;

The ENN decision rule can be approximated as follows:

$$f_{\text{ENN.V2}} = \arg \max_{j \in 1, 2, \dots, N} \{\Delta n_j + k_j - kT_j\} \quad (4.13)$$

where k is the user-defined parameter of the number of the nearest neighbors, Δn_j denotes the number of samples in class j who consider Z as one of their k nearest neighbors, k_j is the number of the nearest neighbors of the test sample Z from class j , and T_j represents the generalized class-wise statistic of original class j (i.e., without the introduction of the test sample Z).

Proof of ENN.V2. Considering two-class classification scenario first, without loss

of generality, if the ENN method classifies Z as class 2 according to Eq. (4.9), we have

$$\begin{aligned} \frac{\Delta n_1^1 + k_1 - kT_1}{(n+1)k} - \frac{\Delta n_2^1}{nk} &< \frac{\Delta n_2^2 + k_2 - kT_2}{(n+1)k} - \frac{\Delta n_1^2}{nk} \\ \Rightarrow \frac{\Delta n_1^1 + \Delta n_1^2 + k_1 - kT_1}{(n+1)k} + \frac{\Delta n_1^2}{(n+1)nk} &< \frac{\Delta n_2^2 + \Delta n_2^1 + k_2 - kT_2}{(n+1)k} + \frac{\Delta n_2^1}{(n+1)nk} \end{aligned} \quad (4.14)$$

When $\frac{\Delta n_1^2}{(n+1)nk}$ and $\frac{\Delta n_2^1}{(n+1)nk}$ approximate to zero, we have

$$\Delta n_1^1 + \Delta n_1^2 + k_1 - kT_1 < \Delta n_2^2 + \Delta n_2^1 + k_2 - kT_2 \quad (4.15)$$

We now define

$$\Delta n_1 = \Delta n_1^1 + \Delta n_1^2 \quad (4.16)$$

$$\Delta n_2 = \Delta n_2^1 + \Delta n_2^2 \quad (4.17)$$

In this way, Δn_1 represents the total number of samples in class 1 who consider Z as one of their k nearest neighbors, and Δn_2 represents the total number of samples in class 2 who consider Z as one of their k nearest neighbors. Therefore, Eq. (4.15) can be expressed as follows:

$$\Delta n_1 + k_1 - kT_1 < \Delta n_2 + k_2 - kT_2 \quad (4.18)$$

Therefore, for a two-class classification problem, the classification rule is

$$f_{\text{ENN.V2}} = \arg \max_{j \in \{1,2\}} \{\Delta n_j + k_j - kT_j\} \quad (4.19)$$

The above derivation can be easily extended to N -class classification problems, and we can obtain the approximate version of our ENN method as shown in Eq. (4.13). $\square$

This approximate version of ENN in Eq. (4.13) also explains why the proposed ENN method can address the scale-sensitive problem in the classic KNN rule. Whereas the KNN rule only considers k_j to make a decision, the proposed ENN method considers three factors to make a prediction decision: two “positive” terms k_j and Δn_j , and one “negative” term kT_j . The two positive terms demonstrate that our ENN approach considers not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors. The negative term means the class that has a greater generalized class-wise statistic would be given a larger penalty value when we estimate the class membership of an unknown sample. The combination of these three factors provides the unique advantage of our ENN method to improve classification performance. We would also like to note that in many practical applications if the two conditions as described in ENN.V2 algorithm are satisfied, using Eq. (4.13) as a simple approximation of our ENN method can in general provide competitive classification performance.

4.1.6 Experiments and Results Analysis

To evaluate the performance of our ENN classifier, we conduct several experiments with Gaussian data classification, hand-written digits classification [88], and twenty real-world datasets from UCI Machine Learning Repository [89]. In all these experiments, we explore every odd number of k from 1 to 21, i.e., $k = 1, 3, 5, 7, \dots, 21$.

We first test our proposed ENN classifier for a 3-dimensional Gaussian data with 3 classes, in comparison with the classic KNN rule and the Maximum a Posterior (MAP) rule. In the MAP rule, we estimate the parameters of Gaussian distribution for each class with the given training data. In our current experiments, we use 250 training samples and 250 test samples for each class with the following

parameters:

$$\boldsymbol{\mu}_1 = [8 \ 5 \ 5]^T \quad \mathbf{C}_1 = \sigma_1^2 \mathbf{I}$$

$$\boldsymbol{\mu}_2 = [5 \ 8 \ 5]^T \quad \mathbf{C}_2 = \sigma_2^2 \mathbf{I}$$

$$\boldsymbol{\mu}_3 = [5 \ 5 \ 8]^T \quad \mathbf{C}_3 = \sigma_3^2 \mathbf{I}$$

Figure 4.3: Overall classification error rate (in percentage) of ENN and KNN for four Gaussian data models.

To demonstrate the effectiveness of the proposed ENN classifier to solve the deficiency of the KNN rule, we examine the classification error rate of each class with different variances. Fig. 4.3 shows the overall classification error rates (in percentage) averaged over 100 random runs. Our results show that the proposed ENN method performs much better than the classic KNN method.

We also evaluate the classification performance of our proposed ENN classifier on the entire MNIST hand-written digits dataset [88], which is a widely used

benchmark in the community. In this experiment, we use 60,000 images as training data and 10,000 images as test data. Fig. 4.4 shows the comparison of classification error rates (in percentage) between classic KNN rule and our ENN rule, where the minimum error of 2.61% is obtained at $k = 7$ for our ENN method. The overall classification error rates using the ENN rule are notably less than those with the KNN rule (Fig. 4.4).

Figure 4.4: Overall classification error rate (in percentage) of ENN and KNN for handwritten digits classification.

We further apply our ENN classifier to twenty real world datasets from UCI Machine Learning Repository [89]. Table 4.1 presents the classification error rates (in percentage) for these twenty UCI datasets in comparison to the classic KNN, naive Bayes, linear discriminant analysis (LDA), and neural network. It shows that ENN always performs better than KNN, and in 17 out of these 20 datasets, the performance improvement is significant (one-tailed t -test, $p = 0.01$). In the neural network implementation, we use the classic multilayer perceptron (MLP) structure with 10 hidden neurons, and with 800 backpropagation iterations for training at the learning rate of 0.01. The results are averaged over 100 random runs, and in every run, we randomly select half of the data as the training data and the remaining half as the test data. Let's consider the Spam database as an example. This database

includes 4601 e-mail messages, in which 2788 are legitimate messages and 1813 are spam messages. Each message is represented by 57 attributes, of which 48 are the frequency of a particular word (FW), 6 are based on the frequency of a particular character (FC), and 3 are continuous attributes that reflect the use of capital letters (SCL) in the e-mails. Fig. 4.5 shows detailed classification error rates (in percentage) with different parameters of k , which clearly demonstrates that ENN method can achieve consistently lower error rates than those of KNN rule.

Figure 4.5: Overall classification error rate (in percentage) of ENN and KNN for spam email classification.

Table 4.1: Average testing error rate and standard derivation of ENN ($k = 3$) compared with KNN ($k = 3$), naive Bayes, LDA, and neural network. All results are shown in percentage. For each dataset, we highlight the best result with **Bold** value among all these five methods. To specifically compare the results of ENN and KNN, we also underline the value if ENN performs significantly better than KNN under one-tailed t -test ($p = 0.01$).

Datasets	ENN	KNN	Naive Bayes	LDA	Neural Network
<i>Ionosphere</i>	<u>17.35</u> ± 2.69	18.55 ± 2.94	19.83 ± 2.86	20.68 ± 3.00	18.48 ± 2.90
<i>Vowel</i>	<u>8.50</u> ± 1.92	11.73 ± 1.80	43.90 ± 2.98	40.94 ± 1.97	45.17 ± 3.15
<i>Sonar</i>	<u>22.67</u> ± 3.97	24.49 ± 4.06	29.22 ± 4.16	33.75 ± 5.11	27.24 ± 4.37
<i>Wine</i>	<u>4.49</u> ± 2.16	7.08 ± 2.20	5.07 ± 1.71	2.58 ± 2.13	7.21 ± 2.88
<i>Breast-cancer</i>	<u>4.04</u> ± 0.87	4.44 ± 1.07	5.76 ± 1.04	5.88 ± 1.18	4.57 ± 1.17
<i>Haberman</i>	31.32 ± 6.53	32.13 ± 5.79	36.35 ± 10.85	34.63 ± 9.92	37.40 ± 10.58
<i>Breast tissue</i>	<u>36.71</u> ± 6.37	42.40 ± 6.19	44.02 ± 6.18	41.24 ± 6.60	67.62 ± 5.22
<i>Movement libras</i>	<u>26.33</u> ± 2.88	32.16 ± 2.97	45.41 ± 3.39	39.90 ± 3.31	40.87 ± 4.34
<i>Mammographic masses</i>	<u>21.16</u> ± 1.43	22.27 ± 1.55	18.96 ± 1.57	19.17 ± 1.72	49.40 ± 0.29
<i>Segmentation</i>	<u>24.71</u> ± 3.07	27.85 ± 3.04	12.64 ± 2.93	12.79 ± 2.88	23.06 ± 5.95
<i>ILPD</i>	40.04 ± 3.58	40.91 ± 3.68	26.87 ± 2.69	29.64 ± 3.39	32.09 ± 3.53
<i>Pima Indians diabetes</i>	<u>31.22</u> ± 2.15	33.08 ± 2.37	29.44 ± 2.19	28.29 ± 2.01	25.38 ± 2.77
<i>Knowledge</i>	<u>23.93</u> ± 4.69	27.11 ± 4.45	12.66 ± 2.45	6.97 ± 2.53	14.42 ± 3.86
<i>Vertebral</i>	<u>35.13</u> ± 4.83	37.64 ± 5.06	47.93 ± 3.41	36.88 ± 4.83	45.11 ± 3.12
<i>Bank note</i>	0.09 ± 0.18	0.12 ± 0.23	15.27 ± 1.25	2.60 ± 0.60	0.18 ± 0.37
<i>Magic</i>	<u>20.10</u> ± 0.33	20.42 ± 0.36	25.69 ± 0.61	23.30 ± 0.34	29.62 ± 0.38
<i>Pen digits</i>	<u>0.74</u> ± 0.15	0.94 ± 0.17	15.38 ± 0.41	11.22 ± 0.52	11.65 ± 0.70
<i>Faults</i>	<u>0.91</u> ± 0.52	1.65 ± 0.86	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
<i>Letter</i>	<u>5.60</u> ± 0.25	7.44 ± 0.25	40.09 ± 0.47	29.80 ± 0.37	28.33 ± 0.52
<i>Spam</i>	<u>10.08</u> ± 0.59	11.52 ± 0.63	10.31 ± 0.78	9.64 ± 0.61	15.32 ± 1.02

We would like to note that for all these experiments, there appears to be no significant difference between ENN and KNN when $k = 1$. The reason for this might be that under such a small value of $k = 1$, the classification performance will be determined by the single closest neighbor. Therefore, when $k = 1$, both methods do not consider the data distribution anyway. That might explain why under $k = 1$, both methods show a very close performance.

4.1.7 Computational Complexity Analysis

As a simple and reliable technique, the nearest neighbor based methods have been widely used in both research and industry for pattern recognition, regression, feature reduction, clustering, among others. However, one major concern of this kind of method is its computational complexity. For the classic KNN method, there is no training stage (i.e., the so-called instance-based learning or lazy learning), with a computational complexity of $O(M \log M)$ in a testing stage for every test sample, where M is the number of training data. To improve the efficiency, it makes more sense to take a preprocessing stage where data structures are built to construct the relationship among training data, such as k-d tree [90], ball tree [91], and nearest feature line [92], or to reduce the size of training data by eliminating the data which may not provide useful information for decision making, such as the condensed nearest neighbor rule by using a concept of mutual nearest neighborhood to only select samples close to the decision boundary [93].

In our ENN method, a preprocessing stage can be developed to calculate the generalized class-wise statistic T_i for each class to build weighted KNN graphs, in which training samples are the vertices and the distances of the sample to its nearest neighbors are the edges. The weighted KNN graphs can then be used to calculate Δn_i^j efficiently for a given test sample. To do so, we calculate distances between the test sample and every training data to obtain k_i for each class at a

computational complexity of $O(M \log M)$. After that, we compare this distance within the weighted KNN graphs to obtain Δn_i^j with only computational complexity of $O(M)$. Therefore, the total computational complexity of our proposed ENN method is $O(M \log M) + O(M)$, which is on the same scale as $O(M \log M)$ in the KNN method. Notice that we have computational complexity of $O(M^2 \log M)$ in a preprocessing stage of building the weighted KNN graphs, if we perform it in a direct manner (i.e., for each training sample, we calculate and order the mutual distances to all the other training data to search its nearest neighbors). Fortunately, the existing techniques proposed in literature to improve the efficiency of KNN based methods can be easily integrated into our ENN method to speed up the searching of nearest neighbors, and hence reduce the complexity of the ENN method. For example, using the structure of k-d trees [90] can reduce the computational complexity of the preprocessing stage and the testing stage to $O(M \log M)$ and $O(\log M)$, respectively.

4.1.8 Discussion

In summary, we develop an innovative ENN classification methodology based on the maximum gain of intra-class coherence. By analyzing the generalized class-wise statistics, ENN is able to learn from the global distribution to improve pattern recognition performance. Unlike the classic KNN rule which only considers the nearest neighbors of a test sample to make a classification decision, ENN method considers not only who are the nearest neighbors of the test sample, but also who consider the test sample as their nearest neighbors. We have developed three versions of the ENN classifier, ENN, ENN.V1 and ENN.V2, and analyzed their foundations and relationships to each other. The experimental results on numerous benchmarks demonstrated the effectiveness of the proposed ENN method.

4.2 A Local Density-Based Approach with Extended Nearest Neighbors for Anomaly Detection

4.2.1 Introduction to Anomaly Detection

Advances in data acquisition have created massive collections of data, capturing valuable information to science, government, business, and society. However, despite of the availability of large amount of data, some events are rare or exceptional, which are usually called “outliers” or “anomalies”. Compared with many other knowledge discovery problems, outlier detection is sometimes more valuable in many applications, such as network intrusion detection, fraudulent transactions, and medical diagnostics. For example, in network intrusion detection, the number of intrusions or attacks (“bad” connections) is much less than the “good” and normal connections. Similarly, the abnormal behaviors are usually rare in many other cases. Although these outliers are only a small portion of the whole data set, it is much more costly to misunderstand them compared with other events.

In recent decades, many outlier detection approaches have been proposed. Usually an outlier detection method can be categorized into the following four types of method [94][95]: distribution-based, distance-based, clustering-based, and density-based. In distribution-based methods, an object is considered as the outlier if it deviates from a standard distribution (e.g., normal, Poisson, etc.) too much [96]. The problem of the distribution-based method is that the underlying distribution is usually unknown and does not follow a standard distribution for many practical applications.

The distance-based methods detect outliers by computing distances among all objects. An object is considered as the outlier when it has d_0 distance away from p_0 percentage of objects in the data set [97]. In [98], the distance among objects is calculated in feature subspace through projections for high dimensional data sets. The problem of these methods is that the local outliers are usually misdetected

for the data set with multiple components or clusters. To detect the local outliers, a top- n k -th nearest neighbor distance is proposed in [99], in which the distance from an object to its k -th nearest neighbor indicates outlierness of the object. The cluster-based methods detect the outlier in the process of finding clusters. The object does not belong any cluster is considered as the outlier [100][101][102].

In density-based methods, an outlier is detected when its local density differs from its neighborhood. Different density estimation methods can be applied to measure the density. In Local Outlier Factor (LOF) [103], an outlierness score, indicating how an object differs from its locally reachable neighborhood, is measured. Previous studies [104][105] have shown that it is more reliable to consider the objects with the highest LOF scores as outliers, instead of comparing the LOF score with a threshold. Several variations of the LOF are also proposed [105][106]. In [105], a Local Distance-based Outlier Factor (LDOF) using the relative distance from an object to its neighbors is proposed for outlier detection in scattered datasets. In [106], a INFLuenced Outlierness (INFLO) score is measured by considering both neighbors and reverse neighbors of an object when estimating its relative density distribution [106]. To address the issue that the LOF method and its variants do not consider the underlying pattern of data, Tang et. al. proposed a connectivity-based outlier factor (COF) scheme in [107]. While the LOF-based and COF-based outlier detection methods use the relative distance to estimate the density, several other density-based methods are proposed based on kernel density estimation [108][109][110]. For example, Local Density Factor (LDF) [108] extends the LOF by using kernel density estimation. In [110], similar to the LOCI, a relative density score termed KDEOS is calculated using kernel density estimation and applies the z -score transformation for score normalization.

We introduce a new outlier detection method based on the local kernel density

estimation for robust local outlier detection. Instead of using the whole data set, the density of an object is estimated with the objects in its neighborhood. Also motivated by the success of our ENN described in §4.1, three kinds of neighbors: k nearest neighbors, reverse nearest neighbors, and shared nearest neighbors, are considered in our local kernel density estimation. A simple but efficient relative density calculation, termed Relative Density-based Outlier Score (RDOS), is introduced to measure the outlierness. Theoretical properties of the RDOS, including the expected value and the false alarm probability are derived, which suggests parameter settings in practical applications. We further employ the top- n scheme to rank the objects with their outlierness, i.e., the objects with the highest RDOS values are considered as the outliers. Simulation results on both synthetic data sets and real-life data sets illustrate superior performance of our proposed method.

In the sequel, we introduce the definition of the RDOS in §4.2.2 and present the detailed descriptions of our proposed outlier detection approach in §4.2.3. In §4.2.4, we derive theoretical properties of the RDOS which suggest the parameter settings of our outlier detection approach. In §4.2.5, we present experimental results and analysis, which show superior performance compared with previous approaches. Finally, discussions are given in §4.2.6.

4.2.2 Local Kernel Density Estimation

We use the KDE method to estimate the density at the location of an object based on the given data set. Given a set of objects $\mathcal{X} = \{X_1, X_2, \dots, X_m\}$, where X_i is the i -th object and $\mathbf{x}_i \in \mathbb{R}^d$ is the corresponding variable for $i = 1, 2, \dots, m$, the KDE method estimates the distribution as follows:

$$p(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \frac{1}{h^d} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right) \quad (4.20)$$

where $K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$ is the defined kernel function with the kernel width of h , which satisfies the following conditions:

$$\int K(u)du = 1, \int uK(u)du = 0, \text{ and } \int u^2K(u)du > 0 \quad (4.21)$$

A commonly used multivariate Gaussian kernel function is given by

$$K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)_{\text{Gaussian}} = \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2h}\right) \quad (4.22)$$

where $\|\mathbf{x}-\mathbf{x}_i\|$ denotes the Euclidean distance between two objects X and X_i . The distribution estimate in Eq. (4.20) offers many nice properties, such as its non-parametric property, continuity and differentiability [111]. Also it is an asymptotic unbiased estimator of the density.

To estimate the density at the location of the object X_p , we only consider its neighbors of X_p as kernels, instead of using all objects in the data set. The reason for this is twofold: firstly, many complex real-life data sets usually have multiple clusters or components, which are the intrinsic patterns of the data. The density estimation using the full data set may lose the local difference in density and fail to detect the local outliers; secondly, the outlier detection will calculate the score for each object, and using the full data set would lead to a high computational cost, which has the complexity of $O(N^2)$ where N is the total number of objects in the data set.

To better estimate the density distribution in the neighbourhood of an object, we propose to use k nearest neighbors, reverse nearest neighbors and shared nearest neighbors as kernels in KDE. Let $NN_r(X_p)$ be the r -th nearest neighbors of the object X_p , we denote the set of k nearest neighbors of X_p as $\mathcal{S}_{KNN}(X_p)$:

$$\mathcal{S}_{KNN}(X_p) = \{NN_1(X_p), NN_2(X_p), \dots, NN_k(X_p)\} \quad (4.23)$$

The *reverse nearest neighbors* of the object X_p are those objects who consider X_p as one of their k nearest neighbors, i.e., X is one reverse nearest neighbor of X_p if

$NN_r(X) = X_p$ for any $r \leq k$. The *shared nearest neighbors* of the object X_p are those objects who share one or more nearest neighbors with X_p , in other words, X is one shared nearest neighbor of X_p if $NN_r(X) = NN_s(X_p)$ for any $r, s \leq k$. We show these three types of nearest neighbors in Fig. 4.6.

Figure 4.6: Three types of nearest neighbors considered. Arrows from X_i and X_j to $NN_r(X_i)$ and $NN_s(X_j)$, respectively.

We denote $\mathcal{S}_{RNN}(X_p)$ and $\mathcal{S}_{SNN}(X_p)$ by the sets of reverse nearest neighbors and shared nearest neighbors of X_p , respectively. For an object, there would be always k nearest neighbors in $\mathcal{S}_{KNN}(X_p)$, while the sets of $\mathcal{RNN}(X_p)$ and $\mathcal{SNN}(X_p)$ can be empty or have one or more elements. Given the three data sets $\mathcal{S}_{KNN}(X_p)$, $\mathcal{S}_{RNN}(X_p)$ and $\mathcal{S}_{SNN}(X_p)$ for the object X_p , we form an extended local neighborhood by combining them together, denoted by $\mathcal{S}(X_p) = \mathcal{S}_{KNN}(X_p) \cup \mathcal{S}_{RNN}(X_p) \cup \mathcal{S}_{SNN}(X_p)$. Thus, the estimated density at the location of X_p is written as

$$p(\mathbf{x}_p) = \frac{1}{|\mathcal{S}(X_p)| + 1} \sum_{X \in \mathcal{S}(X_p) \cup \{X_p\}} \frac{1}{h^d} K\left(\frac{\mathbf{x} - \mathbf{x}_p}{h}\right) \quad (4.24)$$

where $|\mathcal{S}|$ denotes the number of elements in the set of $\mathcal{S}$.

4.2.3 RDOS: Relative Density-based Outlier Factor

After estimating the density at the locations of all objects, we propose a novel relative density-based outlier factor (RDOS) to measure the degree to which the density of the object X_p deviates from its neighborhood, which is defined as follows:

$$RDOS_k(X_p) = \frac{\sum_{X_i \in \mathcal{S}(X_p)} p(\mathbf{x}_i)}{|\mathcal{S}(X_p)| p(\mathbf{x}_p)} \quad (4.25)$$

The RDOS is the ratio of the average neighborhood density to the density of interested object X_p . If $RDOS_k(X_p)$ is much larger than 1, then the object X_p would be outside of a dense cluster, indicating that X_p would be an outlier. If $RDOS_k(X_p)$ is equal or smaller than 1, then the object X_p would be surrounded by the same dense neighbors or by a sparse cloud, indicating that X_p would not be an outlier. In practice, we would like to rank the RDOS values and detect top- n outliers. We summarize our algorithm in Algorithm 4, which takes the KNN graph as input. The KNN graph is a directed graph in which each object is a vertex and is connected to its k nearest neighbors with an outbound direction. In the KNN graph, an object will have k outbound edges to the elements in $\mathcal{S}_{KNN}$, and have none, one or more inbound edges. The KNN graph construction using the brute-force method has the computational complexity of $O(N^2)$ for N objects, and it can be reduced to $O(N \log N)$ using the $k-d$ trees [112]. Using the KNN graph KNN-G, it is easy to obtain the k nearest neighbors $\mathcal{S}_{KNN}$, reverse nearest neighbors $\mathcal{S}_{RNN}$ and shared nearest neighbors $\mathcal{S}_{SNN}$ with an approximate computational cost of $O(N)$. For each object, we form a set of local nearest neighbors $\mathcal{S}$ with the combination of $\mathcal{S}_{KNN}$, $\mathcal{S}_{RNN}$ and $\mathcal{S}_{SNN}$, and calculate the density at the location of the object based on the set of $\mathcal{S}$. Then, we calculate the RDOS value of each object based on the densities of local neighbors in $\mathcal{S}$. The top- n outliers are obtained by sorting the RDOS values in a descending way. If one wants to determine whether an object X_p is outlier, we can compare the value of $RDOS_k(X_p)$ with a threshold τ , i.e., we determine an object is outlier if $RDOS_k(X_p)$ satisfies

$$RDOS_k(X_p) > \tau \quad (4.26)$$

where the threshold τ is usually a constant value that is pre-determined by users.

Algorithm 4: RDOS for top- n outlier detection based on the KNN graph

INPUT: $k, \mathcal{X}, d, h$, the KNN graph KNN-G.

OUTPUT: Top- n objects in $\mathcal{X}$.

ALGORITHM:

```
foreach object  $X_p \in \mathcal{X}$  do
1    $\mathcal{S}_{KNN}(X_p) = \text{getOutboundObjects}(\text{KNN-G}, X_p)$ : get  $k$  nearest
    neighbors of  $X_p$ ;
2    $\mathcal{S}_{RNN}(X_p) = \text{getInboundObjects}(\text{KNN-G}, X_p)$ : get reverse nearest
    neighbors of  $X_p$ ;
3    $\mathcal{S}_{SNN}(X_p) = \emptyset$ : initialize shared nearest neighbors of  $X_p$ ;
4   foreach object  $X \in \mathcal{S}_{KNN}(X_p)$  do
5        $\mathcal{S}_{RNN}(X) = \text{getInboundObjects}(\text{KNN-G}, X)$ ;
6        $\mathcal{S}_{SNN}(X_p) = \mathcal{S}_{SNN}(X_p) \cup \mathcal{S}_{RNN}(X)$ : get objects who share  $X$  as
        nearest neighbors with  $X_p$ ;
    end
7    $\mathcal{S}(X_p) = \mathcal{S}_{KNN}(X_p) \cup \mathcal{S}_{RNN}(X_p) \cup \mathcal{S}_{SNN}(X_p)$ ;
8    $p(X_p) = \text{getKernelDensity}(\mathcal{S}(X_p), X_p, h)$ : estimate the local kernel
    density at the location of  $X_p$ ;
end
foreach object  $X_p \in \mathcal{X}$  do
9   Calculate  $\text{RDOS}_k(X_p)$  for  $X_p$  according to Eq. (4.25);
end
10 Sort RDOS in a descending way and output the top- $n$  objects.
```

4.2.4 Theoretical Properties of RDOS

In this section, we analyze several nice properties of the proposed outlierness metric. In Theorem 1, we give the expected value of RDOS when the object and its neighbors are sampled from the same distribution, which indicates the lower bound of RDOS for outlier detection.

Theorem 2. *Let the object X_p be sampled from a continuous density distribution. For $N \rightarrow \infty$, the RDOS equals 1 with probability 1, i.e., $\text{RDOS}_k(X_p) = 1$, when the kernel function K is nonnegative and integrable.*

Proof. For a fixed k , $N \rightarrow \infty$ indicates that the objects in $\mathcal{S}(X_p)$ locate in the local neighborhood of X_p with the radius $r \rightarrow 0$. Considering data sampled from a continuous density distribution $f(x)$, the expectation of the density estimation

at X_p exists and is consistent to the true one [113]:

$$\mathbb{E}[p(\mathbf{x}_p)] = f(\mathbf{x}_p) \int K(u) du = f(\mathbf{x}_p) \quad (4.27)$$

and its asymptotic variance is given by [113]

$$\mathbb{V}ar[p(\mathbf{x}_p)] = 0 \quad (4.28)$$

Meanwhile, the average density at the neighborhood of X_p with the radius of $r \rightarrow 0$ can be given by

$$\mathbb{E}[\bar{p}(\mathbf{x}_p)] = \mathbb{E}\left[\frac{\sum_{X_i \in \mathcal{S}(X_p)} p(\mathbf{x}_i)}{|\mathcal{S}(X_p)|}\right] = \mathbb{E}[p(\mathbf{x}_p)] = f(\mathbf{x}_p) \quad (4.29)$$

Taking the ratio, we get

$$\mathbb{E}[\bar{p}(\mathbf{x}_p)]/\mathbb{E}[p(\mathbf{x}_p)] = 1 \quad (4.30)$$

□

This theorem shows that when $RDOS_k(X_p) \approx 1$, we could say that the object X_p is not an outlier. Since RDOS is always positive, when $0 < RDOS_k(X_p) < 1$, we could say the object X_p can be ignored in outlier detection. Only these objects whose RDOS values are larger than 1 are possible to be outliers.

We next examine the upper-bound false detection probability to give a sense of threshold selection in practice.

Theorem 3. *Let $\mathcal{S}(X_p)$ be the set of local neighbors of X_p in RDOS, which are assumed to be uniformly distributed in ball B_r centered at X_p with the radius of r . Using the Gaussian kernel, the probability of false detecting X_p as an outlier is given by*

$$P[RDOS_k(X_p) > \gamma] \leq \exp\left(-\frac{2(\gamma-1)^2(|\mathcal{S}|+1)^2(2\pi)^d h^{2d}}{|\mathcal{S}|(2|\mathcal{S}|+\gamma+1)^2 V^2}\right) \quad (4.31)$$

where h is the kernel width and V is the volume of ball B_r .

Proof. For simplicity of notation, we use $\mathcal{S}$ for $\mathcal{S}(X_p)$ and consider $X_p = 0$. Then, the density estimation at X_p given the local neighbors $X_1, X_2, \dots, X_{|\mathcal{S}|}$ is written as

$$p(\mathbf{x}_p) = \frac{1}{|\mathcal{S}| + 1} \sum_{X_i \in \mathcal{S} \cup X_p} \frac{1}{(2\pi)^{d/2} h^d} \exp\left(-\frac{\|\mathbf{x}_i\|^2}{2h}\right) \quad (4.32)$$

and the average density estimation in the neighborhood of X_p is written as

$$\begin{aligned} \bar{p}(\mathbf{x}_p) &= \frac{1}{|\mathcal{S}|} \sum_{X_i \in \mathcal{S}} p(\mathbf{x}_i) \\ &= \frac{1}{|\mathcal{S}|(|\mathcal{S}| + 1)} \sum_{X_i \in \mathcal{S}} \sum_{X_j \in \mathcal{S} \cup X_p} \frac{1}{(2\pi)^{\frac{d}{2}} h^d} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2h}\right) \end{aligned} \quad (4.33)$$

For X_i , $i = 1, 2, \dots, |\mathcal{S}|$, uniformly distributed in ball B_r , we can compute the expectation of both $p(\mathbf{x}_p)$ and $\bar{p}(\mathbf{x}_p)$ from Theorem 1, which is given by:

$$\mathbb{E}[\bar{p}(\mathbf{x}_p)] = \mathbb{E}[p(\mathbf{x}_p)] = \frac{1}{V} = \frac{\pi^{n/2} r^n}{\Gamma(n/2 + 1)} \quad (4.34)$$

where V is the volume of n -sphere B_r and $n = d - 1$. The rest of proof follows the McDiarmid's Inequality which gives the upper bound of the probability that a function of i.i.d. variables $f(X_1, X_2, \dots, X_{|\mathcal{S}|})$ deviates from its expectation. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$, $\forall i, \forall x_1, \dots, x_{|\mathcal{S}|}, x'_i \in \mathcal{S}$,

$$|f(x_1, \dots, x_i, \dots, x_{|\mathcal{S}|}) - f(x_1, \dots, x'_i, \dots, x_{|\mathcal{S}|})| \leq c_i \quad (4.35)$$

Then, for all $\epsilon > 0$,

$$\mathbb{P}[f - \mathbb{E}(f) \geq \epsilon] \leq \exp\left(\frac{-2\epsilon^2}{\sum_{i=1}^{|\mathcal{S}|} c_i^2}\right) \quad (4.36)$$

For $f_1 = p(\mathbf{x}_p)$, we have

$$\begin{aligned} &|f_1(x_1, \dots, x_i, \dots, x_{|\mathcal{S}|}) - f_1(x_1, \dots, x'_i, \dots, x_{|\mathcal{S}|})| \\ &= \frac{K(\mathbf{x}_i/h) - K(\mathbf{x}'_i/h)}{h^d(|\mathcal{S}| + 1)} \leq \frac{1 - \exp(-r^2/2h)}{(2\pi)^{d/2} h^d (|\mathcal{S}| + 1)} = c_1 \end{aligned} \quad (4.37)$$

For $f_2 = \bar{p}(\mathbf{x}_p)$, we have

$$\begin{aligned}
& |f_2(x_1, \dots, x_i, \dots, x_{|\mathcal{S}|}) - f_2(x_1, \dots, x'_i, \dots, x_{|\mathcal{S}|})| \\
&= \frac{K(\frac{\mathbf{x}_i}{h}) - K(\frac{\mathbf{x}'_i}{h}) + 2 \sum_{j=1, j \neq i}^{|\mathcal{S}|} \left[K(\frac{\mathbf{x}_i - \mathbf{x}_j}{h}) - K(\frac{\mathbf{x}'_i - \mathbf{x}_j}{h}) \right]}{h^d(|\mathcal{S}| + 1)} \\
&\leq \frac{1 - \exp(-r^2/2h) + 2|\mathcal{S}|(1 - \exp(-2r^2/h))}{(2\pi)^{d/2}h^d(|\mathcal{S}| + 1)} = c_2
\end{aligned} \tag{4.38}$$

We define a new function $f = f_2 - \gamma f_1$, which is bounded by

$$|f| \leq |f_2| + \gamma |f_1| \leq c_2 + \gamma c_1 \leq \frac{2|\mathcal{S}| + \gamma + 1}{(2\pi)^{d/2}h^d(|\mathcal{S}| + 1)} = c \tag{4.39}$$

Then, the probability of false alarm is written as

$$\mathbb{P}[RDOS_k(X_p) > \gamma] = \mathbb{P}[\bar{p}(\mathbf{x}_p) - \gamma p(\mathbf{x}_p)] = \mathbb{P}[f - \mathbb{E}(f) > t] \tag{4.40}$$

where $t = (\gamma - 1)/V$. From Theorem 1, we are only interested in the case of $RDOS_k(X_p) > 1$, i.e., $\gamma > 1$, and $t > 0$. Using the McDiarmid's Inequality, we have

$$\begin{aligned}
\mathbb{P}[RDOS_k(X_p) > \gamma] &\leq \exp\left(-\frac{2t^2}{\sum_{i=1}^{|\mathcal{S}|} c^2}\right) = \exp\left(-\frac{2t^2}{|\mathcal{S}|c^2}\right) \\
&\leq \exp\left(-\frac{2(\gamma - 1)^2(|\mathcal{S}| + 1)^2(2\pi)^d h^{2d}}{|\mathcal{S}|(2|\mathcal{S}| + \gamma + 1)^2 V^2}\right)
\end{aligned} \tag{4.41}$$

□

4.2.5 Experimental Results and Analysis

We first test the proposed RDOS in two synthetic data sets for outlier detection. Our first synthetic data set includes two Gaussian clusters centered at $(0.5, 0.8)$ and $(2, 0.5)$, respectively, each of which has 100 data samples. There are three outliers around these two clusters, as indicated in Fig. 4.7. To calculate the RDOS, we use $k = 21$ nearest neighbors and $h = 0.01$ in kernel functions. In Fig. 4.8, we show the RDOS of all data samples, where the color and the radius of

circles denote the value of RDOS. It can be shown that the RDOS of these three outliers is significantly larger than that of non-outliers. We further rank data samples in a descending way according to the RDOS, and show the top $n = 5$ data samples with the largest RDOS in Table 4.2.

Figure 4.7: The distribution of normal data and outliers, where the objects: A , B , and C are outliers.

Figure 4.8: The outlierness scores of all data samples, where the value of RDOS is illustrated by the color and the radius of the circle.

The second synthetic data set used in our simulation consists of data samples uniformly distributed around a cosine curve, which can be written as

$$x_2 = \cos(x_1) + w \quad (4.42)$$

where $w \sim \mathcal{N}(0, \sigma^2)$. In our simulation, we use $\sigma^2 = 0.1$, and generate four outliers in this data set, as shown in Fig. 4.9. The RDOS of all data samples is shown

Table 4.2: Top $n = 5$ data samples with the largest RDOS in two synthetic data sets.

Gaussian Data Set			Cosine Data Set		
Rank	Data	RDOS	Rank	Data	RDOS
1	(2.50, 0.20)	22.16	1	(8.80, 0.00)	11.78
2	(1.00, 1.00)	11.68	2	(1.80, 1.00)	11.72
3	(2.00, 0.00)	9.64	3	(5.70, 0.20)	11.68
4	(2.25, 0.59)	3.40	4	(6.50, 0.70)	10.93
5	(0.73, 0.50)	2.31	5	(2.38, -0.45)	2.12

in Fig. 4.10, where both the color and the radius of circles indicate the value of RDOS. It is still shown that the RDOS-based method can effectively detect the outliers. Meanwhile, we show the top $n = 5$ data samples with the largest RDOS in Table 4.2.

Figure 4.9: The distribution of normal data and outliers, where A , B , C and D are considered as outliers.

We also conduct outlier detection experiments on four real-life data sets to demonstrate the effectiveness of our proposed RDOS approach. All of these four data sets are originally from the UCI repository [89], including BREAST CANCER, PEN-LOCAL, PEN-GLOBAL, and SATELLITE, but are modified for local and global outlier detection [114]. We summarize the characteristics of these four data sets in Table 4.3. Before measuring the outlierness, we first normalize the data ranging

Figure 4.10: The outlieriness scores of all data samples, where the value of RDOS is illustrated by the color and the radius of the circle.

from 0 to 1. In Fig. 4.11, we show the first two principle components of these four data sets, where the outliers are denoted by the blue solid circle.

Figure 4.11: The normal data and outliers in four real-life data sets: (A) BREAST CANCER, (B) PEN-LOCAL, (C) PEN-GLOBAL, and (D) SATELLITE. Only the first two principle components are shown.

Since all these data sets are highly imbalanced, the use of overall accuracy is not appropriate. In our experiments, we use the metrics of Receiver Operator Characteristic (ROC) curve and Precision-Recall (PR) curve. The ROC curve shows how the number of correctly classified positive samples (outliers) varies with the number of incorrectly classified negative samples (normal samples). The PR

Table 4.3: The characteristics of four data sets

Dataset	# of features	# of outliers	# of data
BREAST CANCER	30	10	357
PEN-LOCAL	16	10	6714
PEN-GLOBAL	16	90	719
SATELLITE	36	75	5025

curve shows the change of precision with recall which are respectively defined as

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

where TP denotes the number of true positives, FN denotes the number of false negatives, and FP denotes the number of false positives. Compared to the ROC curve, the PR curve can give a more informative picture of an algorithm’s performance particularly when dealing with highly skewed datasets. Both ROC and PR curves can be obtained by comparing the detector output with different thresholds, leading to different pairs of false positive rate and true positive rate in the ROC curve and pairs of precision and recall in the PR curve. For these two types of curves, we use the metric of Area Under the Curve (AUC) to further qualify the detection performance.

We compare our RDOS approach with another four widely used outlier detection approaches: Outlier Detection using Indegree Number (ODIN) [95], LOF [103], INFLO [106], and Mutual Nearest Neighbors (MNN) [102]. Since all of these examined methods are nearest neighbors-based methods, we evaluate the outlier detection performance with different k values. For the proposed approach, we also evaluate its performance with different kernel bandwidth $h = [0.1, 1, 2]$ to illustrate its effect on detection performance.

Table 4.4 shows the detection performance with the metric of AUC of ROC

Table 4.4: Comparison of detection performance with the metric of AUC of ROC curves for four real-life data sets. For each data set, we highlight the best result with **Bold** value and the second one with *Italic* value among all methods.

k	Datasets	ODIN	LOF	INFLO	MNN	RDOS ($h = 0.1$)	RDOS ($h = 1$)	RDOS ($h = 2$)
$k = 7$	BREAST CANCER	0.763	0.967	0.904	0.696	0.941	0.979	<i>0.977</i>
	PEN-LOCAL	0.809	0.982	0.975	0.744	0.990	<i>0.988</i>	<i>0.988</i>
	PEN-GLOBAL	0.531	0.503	0.554	0.418	0.554	0.565	<i>0.557</i>
	SATELLITE	0.639	0.752	<i>0.749</i>	0.603	0.738	0.732	0.731
$k = 11$	BREAST CANCER	0.852	0.976	0.915	0.823	0.955	0.978	<i>0.977</i>
	PEN-LOCAL	0.892	0.990	0.988	0.875	0.990	0.990	0.990
	PEN-GLOBAL	0.593	0.647	0.641	0.501	0.708	0.765	<i>0.761</i>
	SATELLITE	0.671	0.741	0.753	0.612	0.748	0.769	<i>0.768</i>
$k = 15$	BREAST CANCER	0.892	0.977	0.946	0.877	0.957	0.979	0.979
	PEN-LOCAL	0.915	0.989	0.985	0.918	0.994	<i>0.992</i>	<i>0.992</i>
	PEN-GLOBAL	0.656	0.780	0.714	0.577	0.898	0.951	<i>0.945</i>
	SATELLITE	0.687	0.699	0.763	0.609	0.729	<i>0.756</i>	<i>0.756</i>

curves for four data sets, in which the first- and second-best results are highlighted for each data set. It can be shown that the proposed RDOS approaches exhibit superior detection performance with the metric of AUC of ROC. Particularly, the RDOS approach with $h = 1$ and $k = 11$ achieves the best performance for all four data sets. Also, one can see that, apart from the data set of SATELLITE at $k = 7$, all the first- and second-best results are from our three RDOS approaches. Table 4.4 also shows that the performance of AUC of ROC is similar for the proposed RDOS approach when $h = 1$ and $h = 2$, and that $\text{RDOS} > \text{LOF} > \text{INFLO} > \text{ODIN} > \text{MNN}$ is generally true, where the symbol “ $>$ ” means “performs better than”.

Table 4.5 shows the detection performance with the metric of AUC of PR curves for four data sets. Similarly, the first- and second-best results are highlighted in Table 4.5. One can easily see that the proposed three RDOS approaches achieve the first- and second-best performance for all data sets. It can be still shown that the observation of $\text{RDOS} > \text{LOF} > \text{INFLO} > \text{ODIN} > \text{MNN}$ is generally true with the metric of AUC of PR curves.

Table 4.5: Comparison of detection performance with the metric of AUC of PR curves for four real-life data sets. For each data set, we highlight the best result with **Bold** value and the second one with *Italic* value among all methods.

k	Datasets	ODIN	LOF	INFLO	MNN	RDOS ($h = 0.1$)	RDOS ($h = 1$)	RDOS ($h = 2$)
$k = 7$	BREAST CANCER	0.052	<i>0.307</i>	0.235	0.038	0.017	0.179	0.397
	PEN-LOCAL	0.004	0.063	0.040	0.003	0.005	<i>0.124</i>	0.144
	PEN-GLOBAL	0.119	0.145	0.172	0.095	0.122	<i>0.188</i>	0.294
	SATELLITE	0.023	<i>0.150</i>	0.131	0.021	0.019	<i>0.150</i>	0.191
$k = 11$	BREAST CANCER	0.090	0.075	0.211	0.063	0.017	<i>0.219</i>	0.400
	PEN-LOCAL	0.006	0.085	0.060	0.006	0.007	0.185	<i>0.167</i>
	PEN-GLOBAL	0.130	0.226	0.190	0.106	0.190	<i>0.337</i>	0.555
	SATELLITE	0.029	0.117	0.133	0.025	0.019	<i>0.133</i>	0.176
$k = 15$	BREAST CANCER	0.135	0.121	0.145	0.090	0.016	<i>0.235</i>	0.403
	PEN-LOCAL	0.008	0.101	0.065	0.008	0.015	0.188	<i>0.129</i>
	PEN-GLOBAL	0.151	0.352	0.253	0.130	0.410	<i>0.522</i>	0.699
	SATELLITE	0.036	0.114	0.128	0.029	0.019	<i>0.138</i>	0.171

4.2.6 Discussion

A novel local outlier detection method based on local kernel density estimation with “extended” nearest neighbors is developed. Instead of only considering the k nearest neighbors of a data sample, three kinds of neighbors are taken into account: k nearest neighbors, reverse nearest neighbors, and shared nearest neighbors, for local kernel density estimation. A simple but efficient relative density calculation, termed Relative Density-based Outlier Score (RDOS), is introduced to measure the outlierness. The derived theoretical properties of the proposed RDOS measure, including the expected value and the false alarm probability, suggest parameter settings for practical applications. Simulation results on both synthetic data sets and real-life data sets illustrate superior performance of our proposed method.

4.3 Summary

In this chapter, two nearest neighbor based approaches are described for classification and anomaly detection. To address the shortcoming of traditional KNN methods which suffer from the local structure of data, global information and enlarged nearest neighborhoods are considered in these two proposed approaches. As

nearest neighbor-based classification methods are used in many scientific applications because of their easy implementation, non-parametric nature, and competitive classification performance, we would expect the proposed methods and their future variations could have widespread use in many areas of data and information processing, particularly in future intelligent CPS.

CHAPTER 5

High-Performance Distributed Computing Architectures

Next generation CPS require high-performance computing architectures to address the big data analysis issue and new communication architectures to support IoT. This chapter starts by giving a brief introduction of CPS architectures and a review of their related work in §5.1. It then describes a novel hierarchical distributed Fog computing architecture which would be suitable for next generation CPS in §5.2. An implementation of a prototype of the proposed Fog computing architecture is described in §5.3 for smart pipeline.

5.1 Introduction

In CPS, it is essential to build accurate, real-time, and large-scale geospatially distributed sensing networks to monitor physical processes or surrounding environments. The widely distributed sensor networks generate a massive volume of data, which leads to a “Big Data” analysis challenge [115][116]. Meanwhile, the machine-to-machine communication among massive numbers of sensors will dominate future communication network traffic, namely Internet of Things [117][118], instead of traditional Internet of Contents (IoC) in human-to-human and human-to-machine communication. Last but not least, the integrated components require quick feedback controls to ensure the interaction of physical world in multiple temporal and spatial scales. For example, an automatic valve closure is necessary when a rupture along a segment of pipeline is detected, and an optimal distribution of municipal emergency services must be made when a natural disaster is predicted. While the former system requires a neighborhood-level control in seconds, the latter one demands a city-wide decision in hours. Hence, the ubiquitous deployment of sensors in CPS requires a high-performance computing paradigm to support

big data analysis with smart technologies and communications in IoT, providing location-awareness and latency-sensitive computing near the data sources (i.e., at the edge of the network).

Currently, the “pay-as-you-go” Cloud computing paradigm is widely used in enterprises to address the emerging challenges of big data analysis because of its scalable and distributed data management scheme. The rising of Cloud computing and Cloud storage in industry provides a solution to support dynamic scalability in many CPS, such as large scale data management for smart house [119], smart lighting [120] and video surveillance [121], and intensive business and academic computing tasks in education institutions [122]. However, the deployment of massive numbers of sensors in CPS requires location awareness and low latency, which are lacking in current commercial Cloud computing models. To address this issue, one possible solution is to extend or modify current Cloud computing approach. In [123], a volunteer “Sensing and Actuation as a Service” (SAaaS) Cloud computing approach is proposed to ensure that users are able to obtain data from different heterogeneous sensors.

Rather than analyzing all original data from massive sensors, an event-driven platform can greatly reduce the computational burden of cyber-physical systems. In [124], Filipponi et al. introduce an event driven architecture to monitor public areas and infrastructures with heterogeneous sensors. Their architecture consists of two major components: a knowledge processor and a semantic information broker. While the knowledge processor produces notifications of events detected by the sensors, the semantic information broker conveys the notifications to all consumer processors in a publish-and-subscribe manner. The advantage of this architecture is its efficient information broadcasting in a city-wide environment; however, it lacks a quick response mechanism for emergency events that may threaten public

safety.

Fog computing recently, proposed by Cisco, extends the Cloud computing paradigm to run geo-distributed applications throughout the network [30]. In contrast to the Cloud, the Fog not only performs latency-sensitive applications at the edge of network, but also performs latency-tolerant tasks efficiently at powerful computing nodes at the intermediate of network. At the top of the Fog, Cloud computing with data centers can be still used for deep analytics.

In the rest of this chapter, we introduce a hierarchical distributed Fog computing architecture for big data analysis in CPS [125]. Due to the natural characteristic of *geo-distribution* in big data generated by massive sensors, we distribute intelligence at the edge of a layered Fog computing network. The computing nodes at each layer perform latency-sensitive applications and provide quick control loop to ensure the safety of critical infrastructure components. Using smart pipeline monitoring as a use case, we implement a prototypical four-layer Fog-based computing paradigm to demonstrate the effectiveness and the feasibility of the system's city-wide implementation in the future.

5.2 Hierarchical Distributed Fog Computing Architecture

The big data in cyber-physical systems exhibits a new characteristic: *geo-distribution* [126]. This new dimension of big data requires that the data needs to be processed near the sensors at the edge, instead of the data centers in traditional Cloud computing paradigm. Meanwhile, the major computing task of big data analysis in CPS is to identify potential anomalies and hazardous events. It is necessary to offer low latency responses to protect the safety of critical infrastructure components when anomalies and hazardous events are detected. Fog computing is a suitable paradigm by extending the Cloud Computing to the edge of network. Because the data is processed at the edge, quick control loops are feasible using

the Fog computing model.

We show a four-layer Fog computing architecture in Fig. 5.1. At the edge of network, layer 4, is the sensing network which contains numerous sensory nodes. Those sensors are non-invasive, highly reliable, and low cost; thus, they can be widely distributed at various public infrastructures to monitor their condition changes over time. Note that massive sensing data streams are generated from these sensors that are geospatially distributed, which have to be processed as a coherent whole.

Figure 5.1: The 4-layer Fog computing architecture in cyber-physical systems.

The nodes at the edge forward the raw data into the next layer, layer 3, which is comprised of many low-power and high-performance computing nodes or edge devices. Each edge device is connected to and responsible for a local group of sensors that usually cover a neighborhood or a small community, performing data analysis in a timely manner. The output of the edge device has two parts: the first are reports of the results of data processing to an intermediate computing node

at its next upper layer, while the second is simple and quick feedback control to a local infrastructure to respond to isolated and small threats to the monitored infrastructure components.

Layer 2 consists of a number of intermediate computing nodes, each of which is connected to a group of edge devices at layer 3 and associates spatial and temporal data to identify potential hazardous events. Meanwhile, it makes quick response to control the infrastructure when hazardous events are detected. The quick feedback control provided at layers 2 and 3 act as localized “reflex” decisions to avoid potential damage [127]. For example, if one segment of gas pipeline is experiencing a leakage or a fire is detected, these computing nodes will detect the threat and shutdown the gas supply to this section. Meanwhile, all results at these two layers are reported to the top layer, for more complex and large-scaled behavior analysis and condition monitoring.

The top layer is a Cloud Computing data center, providing city-wide monitoring and centralized controlling. Complex, long-term, and city-wide behavior analyses can be also performed at this layer, such as large-scale event detection, long-term pattern recognition, and relationship modeling, to support dynamic decision making. This allows municipalities to perform city-wide response and resource management in the case of a natural disaster or a large-scale service interruption. In summary, the 4-layer Fog computing architecture supports the quick response at neighborhood-wide, community-wide, and city-wide levels.

The hierarchical Fog computing architecture offers high performance computing and communication to address the challenges of big data analysis and provides multilevel response in CPS. It is shown that the workloads of data analysis can be parallelized on massive edge devices and computing nodes. Each edge device or intermediate computing node only performs light-weight computing tasks; thus,

their massive use in parallel offers high-performance computing power for city-wide data analysis. More importantly, such parallel computing mechanisms can easily balance the throughput and load among all edge devices and computing nodes to avoid potential computing bottlenecks.

Figure 5.2: Data and control flows in hierarchical Fog computing architecture.

While the massive parallelization of Fog computing offers high performance computing, its hierarchical architecture also reduces burdens on communication bandwidth. We show the data and control flow among all devices in Fig. 5.2. The massive amount of data is generated from widely distributed sensors at layer 4. Instead of transmitting the raw data to the Cloud, the hierarchical distributed edge devices and intermediate computing nodes at layers 3 and 2 only upload high-level data representation while performing associated computing tasks, which can greatly reduce the data size transmitted to the Cloud. More importantly, these devices can output control signal (as the dashed line signifies) in a timely manner to the monitored infrastructures to ensure their safety when anomalies and hazardous events are detected. At the top layer, a number of computers are connected via the Internet to build Clouds, performing city-level computing tasks and data storage.

5.3 A Prototype of Fog Computing Architecture for Smart Pipeline

In this section, we present the implementation of four-layer Fog computing architecture for smart pipeline monitoring. Pipelines play important role in resource and energy supplying and are essential infrastructure components in cities. However, several threats endanger the integrity of pipeline, such as aging and sudden environmental changes. Those threats lead to corrosion, leakage, and failure of pipelines, with serious economic and ecologic consequences [128] [129]. We show that the hierarchical Fog computing architecture is suitable for accurate and real-time monitoring of city-wide pipelines and provides quick responses when pre-defined threats and hazardous events are detected. Threat pattern recognition and hazardous event detection are performed with advanced machine learning algorithms in computing nodes. The detailed implementation of each individual layer and the employed technologies are described in the rest of this section.

5.3.1 Layer 4: Fiber Optic Sensing Networks

At layer 4, optical fibers are used as sensors to measure the temperature along the pipeline. Optical frequency domain reflectometry (OFDR) system is applied to measure the discontinuity of the regular optical fibers [130]. With the continuous sweep method, the Rayleigh scatter ($\sim -80\text{dB}$) as a function of length along the fiber under test can be obtained via the Fourier transform. With the time-domain filter and cross-correlation method, the extracted frequency patterns at certain locations can be used to detect the ambient physical change, such as strain, stress and temperature [131]. Enhanced sensitivities can be achieved by modifying the regular communication fiber with higher reflectivity [132]. Different cases, such as fire and gas leakage, would introduce the temperature change, which can be monitored by the fiber sensing network.

Figure 5.3: Schematic of the sensor interrogation system (TLS: tunable laser source, CPL: coupler, PC: polarization controller, CIR: circulator, BPD: balanced photodiode, DAQ: data acquisition card)

The Schematic of the OFDR interrogation system is illustrated in Fig. 5.3 [133]. Light from a tunable laser source (TLS) is split into two paths, “clock” and “signal”. “Clock” is an interferometer used to calibrate the nonlinear sweep effect of the TLS by providing a corrected time base for a data acquisition card (DAQ) during frequency sweep. A comparator circuit was designed to receive “clock” signal to trigger the DAQ card to sample “signal” data. Light in the “signal” section is split between the reference and measurement arms of an interferometer via a 50/50 coupler (CPL); in the measurement path, an optical circulator (CIR) further splits the light to interrogate the low-reflection IFPI array and returns the reflected light. A polarization controller (PC) is used to tune the state of polarization in the system. Another 50/50 CPL then recombines the measurement and reference fields. In this setup, the TLS sweeps from 1525 to 1555 nm at a speed of 100 nm/s, covering a total bandwidth of 3.79 THz. The fiber sensors along the pipeline system are connected to the interrogation system for real-time data collection and analysis.

5.3.2 Layer 3: Edge Device for Feature Extraction

Layer 3 is composed of parallelized small computing nodes, or edge devices. Each edge device usually performs two computing tasks. The first task is to identify

potential threat patterns on the incoming data streams from sensors and to output control signal when the threat is detected. Supervised machine learning algorithms, such as neural network, support vector machine, nearest neighbors, and others, can be used to identify pre-defined patterns, and non-supervised machine learning algorithms, such as clustering, can be used to detect data anomalies. Learning models are typically trained offline with experimental data.

The second computing task is to perform feature extraction and to report the results to the intermediate computing nodes at the upper layer for further analysis. Considering a region governed by one edge device with a total length of hundreds of meters, millions of temperature sensors in our high resolution sensing network produce massive data streams and lead to a high data rate. Instead of transmitting the raw sensor data to layer 2, it is necessary to reduce the communication load between the edge devices and the intermediate computing nodes. Thereafter, raw sensor data is discarded.

In our current system implementation, only the second computing task is performed at the edge devices. We refer interested readers to [127] where support vector machine is implemented on FPGA to detect potential threat patterns.

5.3.3 Layer 2: Intermediate Computing Node for Event Recognition with Spatial-Temporal Association

The intermediate computing nodes at layer 2 are connected to tens and hundreds of edge devices, governing the community-level sensors. The data streams from these edge devices represent measurements at various locations. The key is to associate the spatial and temporal data and to identify potential hazardous events.

Assume an intermediate computing node connects n edge devices, and denote a $m \times 1$ vector $\mathbf{s}_i(t)$ by the features outputted from the i -th edge device at time t . Since the sensors are static, the features output from each edge device contains

the geospatial information. After receiving all the features from n edge devices, we combine these n groups of feature vectors into a $mn \times 1$ feature vector $\mathbf{x}(t)$. Hence, from time 1 to time t , this intermediate computing node receives the data sequences $X = \{\mathbf{x}(1), \dots, \mathbf{x}(t)\}$, and the task of event recognition at this layer is to recognize the event pattern given its previous data sequences.

We apply hidden Markov model (HMM) for modeling the spatial-temporal pattern of each event in a probabilistic manner. HMM is a powerful probabilistic tool to represent a dynamic process characterized by a stochastic Markov chain with unobserved or hidden states. From Markov chain theory, the hidden variable moves from one state to another (or to the same state) with a certain probability at each discrete time interval. HMM is a suitable learning model for event recognition, as one can assert that the occurrence of an event is determined by the underlying unobserved variable and different events have different hidden variable transition probabilities.

Two basic procedures are needed in HMM for classification: learning and evaluation. The occurrence of each of the N events is represented by an unique HMM model, denoted by $hmm_i(\mathbf{A}_i, \boldsymbol{\theta}_i)$, $i = 1, 2, \dots, N$. The model parameters $\mathbf{A}_i$ and $\boldsymbol{\theta}_i$ of each HMM is trained offline with the Baum Welch algorithm using the collected experimental data. The Baum Welch learning algorithm is an expectation-maximization (EM) method. At the evaluation stage, the probability of observing a sequence under each individual HMM model is calculated. By applying the maximum a posteriori (MAP) rule, we assign a new observed sequence $\mathbf{x}_{1:t}$ to the i -th event if

$$\begin{aligned} i &= \arg \max_{i=\{1,2,\dots,N\}} p(hmm_i(\mathbf{A}_i, \boldsymbol{\theta}_i) | \mathbf{x}_{1:t}) \\ &= \arg \max_{i=\{1,2,\dots,N\}} p(\mathbf{x}_{1:t} | hmm_i(\mathbf{A}_i, \boldsymbol{\theta}_i)) p(hmm_i(\mathbf{A}_i, \boldsymbol{\theta}_i)) \end{aligned} \quad (5.1)$$

where $p(hmm_i(\mathbf{A}_i, \boldsymbol{\theta}_i))$ is the prior probability of the i -th event. If the occurrence

of each event has a uniform distribution, the above decision rule can be written as

$$i = \arg \max_{i=\{1,2,\dots,N\}} p(\mathbf{x}_{1:t} | \text{hmm}_i(\mathbf{A}_i, \boldsymbol{\theta}_i)) \quad (5.2)$$

We use the efficient forward algorithm to calculate the likelihood of observing the sequence $\mathbf{x}_{1:t}$ under each HMM, where the joint probability of hidden state and observing sequence is recursively updated.

5.3.4 Layer 1: Cloud for Data Management

The top layer is at data centers of the Cloud, which collects data and information from each intermediate computing node on layer 2. We build the Cloud using the open source Hadoop, taking advantage of the power of clusters and high performance computing and storage. The Hadoop framework consists of two major parts: MapReduce and Hadoop Distributed File System (HDFS). It provides automatic parallelization of large scale data analysis workloads and automatic data recovery from one of multiple data copies. Such Cloud models hold many desired properties: *fault tolerance* has *efficiency*. When a machine fails, the data analysis job is transparently assigned to another machine, and the data stored in this failed machine is automatically recovered with its copies from the other machine. For high efficiency, one data analysis task is executed in multiple machines, and it is marked as completed when either of these executions is completed.

5.3.5 Experimental Results and Analysis

In our experiment, we built a prototype of pipeline monitoring system. The layout of pipeline structure is shown in Fig. 5.4. The optical fiber sensors were distributed along this pipeline such that the temperature of pipeline is measured. The real-time data was collected from the fiber sensor network along the prototypical pipeline system. Each sensing element contains a 1cm sensing region, with a temperature resolution of $7.6\text{ }^{\circ}\text{C}$ [134]. The temperature shift was extracted for

each sensing element. The step between the neighboring sensing elements was 1cm and temperature data were gathered along the pipeline system from a total of 205 sensing elements. The system-level update rate was 1 Hz with a color mapping display. In summary, the system had a temporal resolution of 1 second and a spatial resolution of 0.01 meters .

Figure 5.4: Layout of prototype pipeline system.

We simulated multiple events around the pipeline and collected the resulting pipeline temperature sensing data. Each event includes a heating and a cooling process. A heat source was placed nearby, blowing the hot air towards the pipeline system. In each experiment, total 100 frames of data were gathered, where in the first 10 frames the system remained stable, from 11 to 40 frames the heat source was on and from 41 to 100 the heat source was off. One example of a spatial-temporal event pattern is shown in Fig. 5.5, where the 1st, 15th, 40th, and 60th frames are given. In each event the heat source was placed with different angle and distance to generate different patterns, and repeated 10 times with an identical setup. In total, 12 spatial-temporal events were generated for detection.

Features are important for machine learning algorithms. The raw sensor data usually is high-dimensional and redundant. The large number of redundant and irrelevant features would lead to “overfitting” and decrease classification accuracy during testing. Hence, instead of using raw sensor data, two data statistics, the

Figure 5.5: One example of spatial-temporal event pattern. A colormap is used to represent the temperature.

mean and the variance, measuring the central tendency and the dispersion respectively, were extracted as features at each edge device and were reported to the intermediate computing nodes.

Each edge device connects to sensors of a neighborhood. We first assume that 6 edge devices are used to monitor each segment of the pipeline, as shown in Fig. 5.6, and evaluate their recognition performance of the 12 spatial-temporal events. As shown in Fig. 5.6, the intermediate computing node at layer 2 receives 12 spatial features at one time. We train a HMM model for each event. Each HMM model has Q hidden states, and the observation probability distribution is modeled by a Gaussian mixture model (GMM) with K Gaussian components. We perform 10-fold cross validation to evaluate the recognition performance. All the following reported results are averaged over 10 folds. For each test data, we run online inference, i.e., at time frame t , an inference decision is made based on its currently and previously observed sequence $\mathbf{x}_{0:t}$.

The online recognition performance with different number of hidden states is shown in Fig. 5.7a, when $K = 2$ Gaussian components are used in GMM, and the performance with different number of components in GMM is given in Fig. 5.7b, when $Q = 2$ hidden states are used. The results in Fig. 5.7a and Fig. 5.7b illustrate that using more hidden states and Gaussian components in HMM would increase the inference performance due to the growing capacity of HMMs.

However, the complex HMM models need more training data for model parameters estimation and the computational complexity is increased. The results also show that we are able to obtain more than 90% accuracy to classify 12 events at the end of the heating process.

Figure 5.6: Fiber optic sensors governed by 6 edge devices. Each edge device is connected to a neighborhood of sensors.

We next examined the impact of spatial distribution of feature on the inference performance. More specifically, we considered the other four modes of sensor connection to the edge devices, in addition to the one shown in Fig. 5.6 where the sensors of each pipeline segment are connected to one edge device. Denoted the i -th edge device by e_i and the j -th pipeline segment by s_j , the total five modes were configured as followings:

- Mode 1: Six edge devices: $e_1 \leftarrow s_1, e_2 \leftarrow s_2, e_3 \leftarrow s_3, e_4 \leftarrow s_4, e_5 \leftarrow s_5, e_6 \leftarrow s_6$
- Mode 2: Four edge devices: $e_1 \leftarrow [s_1 \ s_2 \ s_5], e_2 \leftarrow s_3, e_3 \leftarrow s_4, e_4 \leftarrow s_6$
- Mode 3: Four edge devices: $e_1 \leftarrow s_1, e_2 \leftarrow [s_2 \ s_3], e_3 \leftarrow s_4, e_4 \leftarrow [s_5 \ s_6]$
- Mode 4: Three edge devices: $e_1 \leftarrow [s_1 \ s_4], e_2 \leftarrow [s_2 \ s_3], e_3 \leftarrow [s_5 \ s_6]$

Figure 5.7: Online inference performance with different number of hidden states in each HMM under different parameter settings: (a). $Q = 2$, $Q = 3$, and $Q = 4$, when two components GMMs are used ($K = 2$); (b). $K = 1$, $K = 2$, and $K = 3$, when two hidden states are used ($Q = 2$).

- Mode 5: One edge device: $e_1 \leftarrow [s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6]$

Figure 5.8: Online inference performance for different sensor connections to the edge devices.

The averaged online inference performance for these five modes is shown in Fig. 5.8, when $Q = 2$ and $K = 2$. The performance comparison among these five modes shows that the performance decreases when less edge devices are used. For the pipeline with fixed length, less edge devices mean that some of them would cover more sensors. If only the global data statistics, such as the mean and

variance, are extracted as features, the spatial resolution of observation in HMM modeling would be reduced. Thus, if the sensors connected to one edge device covers a long pipeline, this result suggests that it is necessary to segment sensing data and extract the statistics of each segment as features to improve the spatial resolution for spatial-temporal learning. However, it is worth noting that high spatial resolution also leads to feature irrelevance and a heavier computational burden. For example, one can get the maximum spatial resolution if the raw sensing data is used for spatial-temporal learning.

5.3.6 Discussion

The Fog computing architecture has significant advantages over the Cloud computing architecture for intelligent CPS. First, the distributed computing and storage nodes of Fog computing ideally suited to support the massive numbers of sensors distributed throughout a city to monitor infrastructure and environmental parameters. If Cloud computing alone is used for this task, huge amounts of data will need to be transmitted to data centers, necessitating massive communication bandwidth and power consumption. Specifically, suppose that we use current sensing setup with 1cm spatial-resolution and 0.5s time-resolution, and that each edge device covers 10m pipeline and each computing node connects 5 edge devices. Considering the total pipeline length L ranging from 10km to 50km, we compare the size of data that needs to be sent to the Cloud per second in Fig. 5.9a for the following three cases: our current Fog computing architecture with layers 2 and 3, the Fog computing architecture with only layer 3 by removing the computing tasks at layer 2 to the Cloud, and the traditional Cloud computing architecture in which both computing tasks at layers 2 and 3 are executed at Cloud. To clearly illustrate the difference among these three architectures, we plot log values of data size. The results in Fig. 5.9a show that using Fog computing, the data transmitted is about

0.02% of the total size, significantly reducing transmission bandwidth and power consumption.

Figure 5.9: Fog vs Cloud computing: (a). The amount of data that are sent to the Cloud per second; (b). The response time for hazardous events, when the Internet bandwidth is $1Mb/s$; (c) The response time for hazardous events with different Internet bandwidths. The log values in the y -axis are used to clearly illustrate the comparisons.

Second, Fog computing supports real-time interactions. Because of high burdens on data transmission, Cloud computing fails to provide real-time control. To quantify the response time for hazardous events under the above three computing architectures, we assume that the execution speed in computing node is 1GIPS, and we omit the memory access time for simplifying our analysis. The comparison of response time for these three architectures is shown in Fig. 5.9b, when the Internet bandwidth connecting to the Cloud is $1Mb/s$. It is seen that the response time is dominated by the data transmission in Cloud computing. Fig. 5.9c also shows the response time when different Internet bandwidths are considered.

As shown in Fig. 5.1, different levels of latency of response can be provided in the Fog computing, which is distinct from the batch processing of Cloud computing. These results illustrate that Fog computing addresses the big data analysis challenge by distributing computing tasks to the edge devices and computing nodes at the edge of network, thus offering optimal responses to changes in city environment.

5.4 Summary

This chapter introduces a hierarchical distributed Fog computing architecture in CPS. In contrast to the Cloud computing, the Fog computing parallelizes data processing at the edge of network, which satisfies the requirements of location awareness and low latency. Moreover, the proposed Fog computing can not only support the Internet of Things, but also offer multiscale coverage ranging from components, to neighbors, to communities and to cities and multiscale latency ranging from milliseconds, to seconds, to minutes and to hours. Therefore, the proposed multi-layer Fog computing architecture is able to support quick response at neighborhood-wide, community-wide and city-wide levels, providing high computing performance and intelligence in future CPS. The “smartness” of physical systems can be enhanced by employing advanced machine learning algorithms across all system layers.

CHAPTER 6

Application: False Data Injection Detection in Smart Grid

Smart energy system, or termed smart grid, is one of the appealing CPS, which incorporates central power plants, energy storage, transmission lines, and distributed energy management units in smart buildings. Compared with the traditional centralized electricity grid, the smart energy system becomes less centralized but more distributed and more transparent to the customers. This transparency introduces the security issues in smart grid as attackers can also easily access the network. This chapter briefly introduces the security issues of smart grid and summarizes our contribution in §6.1. It then describes background and related works in §6.2, and proposes new statistical estimation and detection methods in §6.3. Simulation results and analysis are given in §6.4.

6.1 Introduction

The power system relies on accurate measurements of system topology and state variables to analyze real-time system dynamics and maintain stable operation. However, measurements collected by the supervisory control and data acquisition (SCADA) system are often corrupted by random noises or missing data [135]. To recover accurate system variables and detect potential bad data for system operation, robust state estimation techniques are commonly used in the energy management systems (EMS) [136].

However, as the smart grid brings in cybernetic integration with the computerized communication network to the modern electrical power infrastructure, the industry and research community have witnessed growing security concerns from false data injection (FDI) attacks in state estimation. Due to the linear approximation in state estimation, malicious attackers can construct stealth schemes to

inject random or targeted false data into the power system measurements that can result in serious instability in system operation.

Although numerous studies have investigated the mathematical methods to build robust state estimation and detection mechanisms against the FDI attacks, they all are built on a common assumption that the background noise in state estimation are white Gaussian noise (WGN) [137, 138, 139]. However, many natural phenomena, such as ice cracking and atmospheric noise, and man-made noise sources, such as electronic devices, can be modeled more accurately as non-Gaussian distributions [140][141]. Detection performance of conventional state estimator and false data detector would deteriorate due to the presence of colored Gaussian noise or even more complex non-Gaussian noises.

In this chapter, we investigate the problems of state estimation and false data injection detection when the measurements are corrupted by the colored Gaussian noise. We model the colored Gaussian noise via the autoregressive (AR) process, and derive a closed form of state estimation and a generalized likelihood ratio test (GLRT) detector for false data injection detection. We show the deterioration of conventional WGN-based bad data detector in colored Gaussian noise and propose an AR-based detector. Moreover, it can be shown that the conventional Gaussian noise approach is the special case of the proposed AR approach. The computer simulations on the IEEE 30-bus power system are conducted to evaluate the detection performance of the proposed AR approach.

6.2 Background and Related Work

State estimations are performed on power transmission networks which consist of a set of generators, load buses, transmission lines, and other electrical facilities governed by physical laws. Measurements are collected on meters, e.g., voltage meters and phasor measurement units (PMU), and reported to wide-area control

centers through the SCADA system. The state variables of the power system is further estimated at the control centers from these measurements with the knowledge of power system topology. Control actions are then determined to maintain stable, cooperative transmission from power plants to customers in interconnected power grids.

Traditional state estimators are capable of identifying and eliminating bad data from state estimations [142][143]. However, it has been recently found that malicious data attacks can exploit system topology information to construct false data injection schemes that will bypass bad data detectors in conventional state estimation [144][145][146]. A number of defending strategies have been proposed [147, 148, 149]. Studies also proposed optimal PMU placement to detect FDIs in the smart grid. Recently, machine learning algorithms have also been introduced to detect stealth phases [150]. These studies, however, are based on the assumption of white Gaussian noise embedded in the measurements. Their performance remains unknown in the presence of a colored-Gaussian or non-Gaussian noise. In this paper, an AR detector is proposed and tested for temporally colored Gaussian noise in state estimators to detect random false data injection in the smart grid.

State Estimation: The state estimation is first proposed by F. Schweppe in 1970[151, 152, 153] as a weighted least-squares (WLS) problem. It is since enriched by numerous studies in the following decades [154, 155, 156]. For a power system with K state variables $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_K]^T$, we have M meter measurements $\mathbf{x} = [x_1, x_2, \dots, x_M]^T$, which is given by

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w} \quad (6.1)$$

where $\mathbf{H}$ is a $M \times K$ ($M > K$) Jacobian topological matrix and $\mathbf{w}$ is a $M \times 1$ measurement error (noise) vector. The state variables typically include the amplitudes and the phases of voltages in buses. Commonly, the measurement error is

modeled by the white Gaussian distribution, i.e., $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_w)$. For this case, it is well-known that the estimated state $\hat{\boldsymbol{\theta}}$ can be given by the same solution using maximum likelihood estimation (MLE) and weighted least squares (WLS) [157], as follows:

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T \mathbf{\Sigma}_w^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{\Sigma}_w^{-1} \mathbf{x} \quad (6.2)$$

Detection of False Data Injection: False data injection has been recently identified as a critical type of malicious data attacks in a power system [158, 159, 160]. It is necessary to detect the false data injection to protect the safety and the integrity of the power system. Technically, the false data injection can be modeled as follows:

$$\mathbf{x} = \mathbf{H}\boldsymbol{\theta} + \mathbf{a} + \mathbf{w} \quad (6.3)$$

where $\mathbf{a}$ is the false data injected to the measurements. $\mathbf{a}$ is usually a sparse vector due to the fact that the attacker can only get access to a limited number of component measurements in power system. We call a vector $\mathbf{a}$ has sparsity of d if there are at most D non-zero elements in the vector, i.e., $\|\mathbf{a}\|_0 = D$. For power engineers, both the state variables $\boldsymbol{\theta}$ and the false data $\mathbf{a}$ in Eq. (6.3) are unknown. Note that the false data with the form of $\mathbf{a} = \mathbf{H}\boldsymbol{\theta}_a$ cannot be detected without knowing prior knowledge of state variables, commonly termed unobservable false data. Attackers may fabricate such attacks by gaining intelligence on the system topology $\mathbf{H}$. Mathematically, the following theorem further indicates whether a false data is observable:

Theorem 4. *For a given measurement matrix $\mathbf{H}$ with a size of $M \times K$, where $M > K$, there always exists a $M \times (M - K)$ matrix $\mathbf{B}$ such that the columns of $\mathbf{B}$ span the orthogonal subspace of the columns of $\mathbf{H}$, i.e., $\mathbf{B}^T \mathbf{H} = \mathbf{0}$ and $\mathbf{B}^T \mathbf{B} = \mathbf{I}$.*

Then any false data $\mathbf{a} \in \mathcal{R}^M$ can be written as: $\mathbf{a} = \mathbf{H}\boldsymbol{\theta}_a + \mathbf{B}\boldsymbol{\theta}_b$, where $\boldsymbol{\theta}_a \in \mathcal{R}^K$ and $\boldsymbol{\theta}_b \in \mathcal{R}^{M-K}$.

The proof of Theorem 4 follows the Orthogonal Decomposition Theorem [161] directly. From Theorem 4, it can be shown that the false data is observable when $\boldsymbol{\theta}_b \neq \mathbf{0}$, and $\boldsymbol{\theta}_a$ is part of state which is estimated as $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta} + \boldsymbol{\theta}_a$. Defining a new state variable as $\boldsymbol{\theta}_1 = \boldsymbol{\theta} + \boldsymbol{\theta}_a$, the measurements can be further written as: $\mathbf{x} = \mathbf{H}\boldsymbol{\theta}_1 + \mathbf{B}\boldsymbol{\theta}_b + \mathbf{w}$, and the hypothesis testing problem for the false data injection detection becomes:

$$\begin{aligned}\mathcal{H}_0 : \boldsymbol{\theta}_b &= \mathbf{0} \\ \mathcal{H}_1 : \boldsymbol{\theta}_b &\neq \mathbf{0}\end{aligned}\tag{6.4}$$

For the measurements corrupted by the noise $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we give the following GLRT detector for the hypothesis testing problem in Eq. (6.4):

Theorem 5. *For the given measurement $\mathbf{x} = \mathbf{H}\boldsymbol{\theta}_1 + \mathbf{B}\boldsymbol{\theta}_b + \mathbf{w}$, where both $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_b$ are unknown, $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{B}^T\mathbf{H} = \mathbf{0}$ and $\mathbf{B}^T\mathbf{B} = \mathbf{I}$, the GLRT detector for the hypothesis testing problem in Eq. (6.4) is to decide $\mathcal{H}_1$ if*

$$T(\mathbf{x}) = 2 \ln \frac{p(\mathbf{x}; \hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_b)}{p(\mathbf{x}; \hat{\boldsymbol{\theta}}_1, \mathbf{0})} = \mathbf{x}^T \mathbf{P}_{\mathbf{H}}^\perp \mathbf{x} > \tau\tag{6.5}$$

where τ is the threshold, $\mathbf{P}_{\mathbf{H}}^\perp = \mathbf{I} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$, and $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_b$ are given by

$$\begin{aligned}\hat{\boldsymbol{\theta}}_1 &= (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x} \\ \hat{\boldsymbol{\theta}}_b &= \mathbf{B}^T\mathbf{x}\end{aligned}\tag{6.6}$$

Proof. Since $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we have $\mathbf{x} \sim \mathcal{N}(\mathbf{H}\boldsymbol{\theta}_1 + \mathbf{B}\boldsymbol{\theta}_b, \mathbf{I})$. The log-likelihood of $\mathbf{x}$ is given by

$$J(\boldsymbol{\theta}_1, \boldsymbol{\theta}_b) = -\frac{1}{2}(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}_1 + \mathbf{B}\boldsymbol{\theta}_b)^T(\mathbf{x} - \mathbf{H}\boldsymbol{\theta}_1 + \mathbf{B}\boldsymbol{\theta}_b) + c\tag{6.7}$$

where c is a constant.

Using the maximum likelihood estimation criterion, both $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_b$ can be estimated as follows

$$\begin{aligned}\frac{\partial J(\boldsymbol{\theta}_1, \boldsymbol{\theta}_b)}{\partial \boldsymbol{\theta}_1} &= 0 \\ \frac{\partial J(\boldsymbol{\theta}_1, \boldsymbol{\theta}_b)}{\partial \boldsymbol{\theta}_b} &= 0\end{aligned}\tag{6.8}$$

which leads to

$$\begin{aligned}\hat{\boldsymbol{\theta}}_1 &= (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \\ \hat{\boldsymbol{\theta}}_b &= \mathbf{B}^T \mathbf{x}\end{aligned}\tag{6.9}$$

The GLRT detector can be written as

$$T(\mathbf{x}) = 2 \ln \frac{p(\mathbf{x}; \hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_b)}{p(\mathbf{x}; \hat{\boldsymbol{\theta}}_1, \mathbf{0})} = \mathbf{x}^T (\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T) \mathbf{x} > \tau\tag{6.10}$$

where τ is the threshold. If $T(\mathbf{x}) > \tau$, then we accept $\mathcal{H}_1$, otherwise we accept $\mathcal{H}_0$. $\square$

To apply Theorem 5 for false data injection detection with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_w)$, we first define a pre-whitened variable $\mathbf{y} = \mathbf{M}\mathbf{x}$ where $\mathbf{M}$ is the known whitening transformation matrix such that $\mathbf{M}^T \mathbf{M} = \boldsymbol{\Sigma}_w^{-1}$, and $\mathbf{H}' = \mathbf{M}\mathbf{H}$. We then have $\mathbf{y} = \mathbf{H}'\boldsymbol{\theta} + \mathbf{a}' + \mathbf{w}'$, where $\mathbf{a}' = \mathbf{M}\mathbf{a}$ and $\mathbf{w}' = \mathbf{M}\mathbf{w}$. Using the pre-whitening transformation, we have $\mathbf{w}' \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. According to the Theorem 4, there always exists a matrix $\mathbf{B}'$ such that $\mathbf{B}'^T \mathbf{H}' = \mathbf{0}$ and $\mathbf{B}'^T \mathbf{B}' = \mathbf{I}$, and $\mathbf{a}' = \mathbf{H}'\boldsymbol{\theta}_a + \mathbf{B}'\boldsymbol{\theta}_b$. Thus, we have $\mathbf{y} = \mathbf{H}'(\boldsymbol{\theta} + \boldsymbol{\theta}_a) + \mathbf{B}'\boldsymbol{\theta}_b + \mathbf{w}'$. According to the Theorem 5, hence, we have the following GLRT detector for the false data injection detection:

$$\begin{aligned}T(\mathbf{y}) &= \mathbf{y}^T (\mathbf{I} - \mathbf{H}'(\mathbf{H}'^T \mathbf{H}')^{-1} \mathbf{H}'^T) \mathbf{y} \\ &= \mathbf{x}^T \mathbf{M}^T (\mathbf{I} - \mathbf{H}'(\mathbf{H}'^T \mathbf{H}')^{-1} \mathbf{H}'^T) \mathbf{M} \mathbf{x} \\ &= \mathbf{x}^T (\mathbf{I} - \boldsymbol{\Sigma}_w^{-1} \mathbf{H}(\mathbf{H}^T \boldsymbol{\Sigma}_w^{-1} \mathbf{H})^{-1} \mathbf{H}^T) \boldsymbol{\Sigma}_w^{-1} \mathbf{x}\end{aligned}\tag{6.11}$$

Sequential Observations: In FDI attacks, an adversary usually hacks the meter measurements for a period of time to mislead the decision making in power systems. The Cramer-Rao Lower Bound (CRLB) theorem indicates that using multiple observations leads to a much lower variance of the state estimation. Given N sequential observations, the estimated state in power system can be written as:

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T \boldsymbol{\Sigma}_w^{-1} \mathbf{H})^{-1} \mathbf{H}^T \boldsymbol{\Sigma}_w^{-1} \bar{\mathbf{x}} \quad (6.12)$$

where $\bar{\mathbf{x}} = \sum_{i=1}^N \mathbf{x}_i / N$ is the mean of N observations, and the GLRT detector for an unknown false data can be given by

$$T(\mathbf{X}) = \frac{2}{N} \ln \frac{p(\mathbf{X}; \hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_b)}{p(\mathbf{X}; \hat{\boldsymbol{\theta}}_1, \mathbf{0})} = \bar{\mathbf{x}}^T (\mathbf{I} - \boldsymbol{\Sigma}_w^{-1} \mathbf{H} (\mathbf{H}^T \boldsymbol{\Sigma}_w^{-1} \mathbf{H})^{-1} \mathbf{H}^T) \boldsymbol{\Sigma}_w^{-1} \bar{\mathbf{x}} \quad (6.13)$$

Here, we assume that the false data is a targeted measurement vector that the attacker intends to inject over the N observations. It can be further extended to state-dependent or random but disruptive vectors under complex attack schemes. The variation of states is also negligible during the period of N observations. As a simplification of complex power system dynamics, this steady-state assumption is commonly used in many studies on DC state estimation and contingency analysis [162][163]. To further validate this assumption, we provide more simulation results in the Supplemental Material with dynamic loading change in the system.

6.3 Proposed State Estimator and False Data Detector with Colored Gaussian Noise

We consider the problems of both state estimation and false data detection in power systems when the measurements are corrupted by the colored Gaussian noise which is modeled by an AR process, and the conventional estimator and detector with the Gaussian noise can be considered as the special case of our methodology. The observation matrix $\mathbf{X}$ can be rewritten as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]^T$, and the measurement error (noise) matrix $\mathbf{W}$ can be rewritten as $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]^T$,

where both $\mathbf{x}_i$ and $\mathbf{w}_i$ are $N \times 1$ vectors. For a power system without false data injection, we have

$$\mathbf{x}_i = \mathbf{1}_N \mathbf{h}_i^T \boldsymbol{\theta} + \mathbf{w}_i, \quad i = 1, 2, \dots, M \quad (6.14)$$

where $\mathbf{h}_i^T$ is the i -th row of $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_M]^T$, and $\mathbf{1}_N$ is a $N \times 1$ all-ones vector. Unlike the existing approaches with the assumption of white Gaussian noise, we consider that the sequential noise $\mathbf{w}_i = [w_{i,0}, w_{i,1}, \dots, w_{i,N}]^T$ for the i -th meter measurement follows a colored Gaussian distribution which is modeled via a p -order AR process:

$$w_{i,n} = \sum_{j=1}^p \alpha_{i,j} w_{i,n-j} + v_{i,n}, \quad n = 0, 1, \dots, N-1 \quad (6.15)$$

where $\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,p}$ are the parameters of the AR process, and $v_{i,n}$ is an independent and identically distributed (I.I.D.) random variable which satisfies a white Gaussian distribution, i.e., $v_{i,n} \sim \mathcal{N}(0, \sigma_i^2)$. It is known that $w_{i,n}$ and $\mathbf{w}_i$ are also Gaussian. Given $w_{i,-1}, w_{i,-2}, \dots, w_{i,-p}$, it can be shown that the N sequential observations of the i -th measurement have a probability density function (PDF) $p(\mathbf{x}_i; \boldsymbol{\theta})$ [164] as follows:

$$\begin{aligned} p(\mathbf{x}_i; \boldsymbol{\theta}) &= \prod_{n=0}^{N-1} p(x_{i,n} | x_{i,n-1}, x_{i,n-2}, \dots, x_{i,n-p}) \\ &= \frac{1}{(2\pi\sigma_i^2)^{N/2}} \exp \left\{ -\frac{1}{2\sigma_i^2} \sum_{n=0}^{N-1} \left[w_{i,n} - \sum_{j=1}^p \alpha_{i,j} w_{i,n-j} \right]^2 \right\} \end{aligned} \quad (6.16)$$

where the exponential term in above equation can be rewritten as

$$\begin{aligned} \sum_{n=0}^{N-1} \left[w_{i,n} - \sum_{j=1}^p \alpha_{i,j} w_{i,n-j} \right]^2 &= (\mathbf{T}_i \mathbf{w}_i + \mathbf{c})^T (\mathbf{T}_i \mathbf{w}_i + \mathbf{c}) \\ &= (\mathbf{T}_i \mathbf{x}_i + \mathbf{c}_i - \mathbf{T}_i \mathbf{1}_N \mathbf{h}_i^T \boldsymbol{\theta})^T (\mathbf{T}_i \mathbf{x}_i + \mathbf{c}_i - \mathbf{T}_i \mathbf{1}_N \mathbf{h}_i^T \boldsymbol{\theta}) \end{aligned} \quad (6.17)$$

where $\mathbf{T}_i$ is given by

$$\mathbf{T}_i = \begin{bmatrix} \alpha_{i,0} & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ -\alpha_{i,1} & \alpha_{i,0} & 0 & \cdots & \cdots & \cdots & 0 \\ & \ddots & & & & & \vdots \\ -\alpha_{i,p} & -\alpha_{i,p-1} & \cdots & \alpha_0 & \cdots & \cdots & 0 \\ 0 & -\alpha_{i,p} & -\alpha_{i,p-1} & \cdots & \alpha_{i,0} & \cdots & 0 \\ & \ddots & & & & & \vdots \\ 0 & \cdots & \cdots & -\alpha_{i,p} & -\alpha_{i,p-1} & \cdots & \alpha_{i,0} \end{bmatrix} \quad (6.18)$$

and $\mathbf{c}_i$ is given by

$$\mathbf{c}_i = \left[-\sum_{k=1}^p \alpha_{i,k}x[-k], -\sum_{k=2}^p \alpha_{i,k}x[-k], \dots, -\alpha_{i,p}x[-p], 0, \dots, 0 \right]^T \quad (6.19)$$

Therefore, it can be shown that $\mathbf{T}_i \mathbf{x}_i + \mathbf{c}_i \sim \mathcal{N}(\mathbf{T}_i \mathbf{1}_N \mathbf{h}_i^T \boldsymbol{\theta}, \sigma_i^2 \mathbf{I})$. For all M measurements, we have the PDF $p(\mathbf{X}; \boldsymbol{\theta})$ as follows:

$$p(\mathbf{X}; \boldsymbol{\theta}) = \prod_{i=1}^M p(\mathbf{x}_i; \boldsymbol{\theta}) \quad (6.20)$$

and the log-likelihood $J(\boldsymbol{\theta})$ as follows:

$$J(\boldsymbol{\theta}) = \log p(\mathbf{X}; \boldsymbol{\theta}) = -\sum_{i=1}^M \frac{1}{2\sigma_i^2} (\mathbf{T}_i \mathbf{x}_i + \mathbf{c}_i - \mathbf{T}_i \mathbf{1}_N \mathbf{h}_i^T \boldsymbol{\theta})^T (\mathbf{T}_i \mathbf{x}_i + \mathbf{c}_i - \mathbf{T}_i \mathbf{1}_N \mathbf{h}_i^T \boldsymbol{\theta}) \quad (6.21)$$

State Estimation: For a power system without false data injection, using the maximum likelihood estimation criterion, we have the following the state estimation solution

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta} \in \Theta} J(\boldsymbol{\theta}) = \left(\sum_{i=1}^M \frac{\mathbf{1}_N^T \mathbf{T}_i^T \mathbf{T}_i \mathbf{1}_N}{\sigma_i^2} \mathbf{h}_i \mathbf{h}_i^T \right)^{-1} \left(\sum_{i=1}^M \frac{\mathbf{1}_N^T \mathbf{T}_i^T (\mathbf{T}_i \mathbf{x}_i + \mathbf{c}_i)}{\sigma_i^2} \mathbf{h}_i \right) \quad (6.22)$$

Let $\mathbf{A} = \text{diag}(a_1/\sigma_1^2, a_2/\sigma_2^2, \dots, a_M/\sigma_M^2)$ be a $M \times M$ diagonal matrix, where $a_i = \mathbf{1}_N^T \mathbf{T}_i^T \mathbf{T}_i \mathbf{1}_N$ for $i = 1, 2, \dots, M$, and a $M \times 1$ vector $\mathbf{z} = [z_1/\sigma_1^2, z_2/\sigma_2^2, \dots, z_M/\sigma_M^2]^T$, where $z_i = \mathbf{1}_N^T \mathbf{T}_i^T (\mathbf{T}_i \mathbf{x}_i + \mathbf{c}_i)$ for $i = 1, 2, \dots, M$, Eq. (6.22) can be rewritten as

$$\hat{\boldsymbol{\theta}} = (\mathbf{H}^T \mathbf{A} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{z} \quad (6.23)$$

Detection of False Data Injection: Following the Theorem 4 and 5, we first define a pre-whitened variable $\mathbf{y} = \mathbf{M} \mathbf{x}$, where $\mathbf{M}^T \mathbf{M} = \mathbf{A}$, and define $\mathbf{H}' = \mathbf{M} \mathbf{H} =$

$[\mathbf{h}_1', \dots, \mathbf{h}_M']^T$. Since $\mathbf{A}$ is a diagonal matrix, $\mathbf{M}$ is also a diagonal matrix and $\mathbf{M} = \text{diag}(\sqrt{a_1}/\sigma_1, \dots, \sqrt{a_M}/\sigma_M)$. According to the Theorem 4, the i -th measurement with false data injection can be written as $\mathbf{y}_i = \mathbf{1}_N \mathbf{h}_i'^T \boldsymbol{\theta}_1 + \mathbf{1}_N \mathbf{b}_i'^T \boldsymbol{\theta}_b + \mathbf{w}_i'$, where $\mathbf{b}_i'^T \mathbf{h}_i' = 0$ and $\mathbf{b}_i'^T \mathbf{b}_i' = 1$, and $\mathbf{w}_i'$ is still modeled by the AR process as follows:

$$w_{i,n}' = \sum_{j=1}^p \alpha_{i,j} w_{i,n-j}' + v_{i,n}', \quad n = 0, 1, \dots, N-1 \quad (6.24)$$

where $v_{i,n}' \sim \mathcal{N}(0, a_i)$. Hence, it can be shown that $\mathbf{T}_i \mathbf{y}_i + \mathbf{c}_i \sim \mathcal{N}(\mathbf{T}_i \mathbf{1}_N \mathbf{h}_i'^T \boldsymbol{\theta}_1 + \mathbf{T}_i \mathbf{1}_N \mathbf{b}_i'^T \boldsymbol{\theta}_b, a_i \mathbf{I})$. The MLEs of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_b$ can be given by

$$\hat{\boldsymbol{\theta}}_1 = (\mathbf{H}'^T \mathbf{H}')^{-1} \mathbf{H}'^T \mathbf{z}' \quad (6.25)$$

$$\hat{\boldsymbol{\theta}}_b = \mathbf{B}'^T \mathbf{z}' \quad (6.26)$$

where $\mathbf{z}' = [z_1'/a_1, \dots, z_M'/a_M]^T$ and $z_i' = \mathbf{1}_N^T \mathbf{T}_i^T (\mathbf{T}_i \mathbf{y}_i + \mathbf{c}_i)$ for $i = 1, 2, \dots, M$. Hence, for the hypothesis testing problem in Eq. (6.4), we have the following GLRT detector for the false data injection detection:

$$T(\mathbf{Y}) = \frac{2}{N} \ln \frac{p(\mathbf{Y}; \hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_b)}{p(\mathbf{Y}; \hat{\boldsymbol{\theta}}_1, \mathbf{0})} = \frac{1}{N} \mathbf{z}'^T (\mathbf{I} - \mathbf{H}' (\mathbf{H}'^T \mathbf{H}')^{-1} \mathbf{H}'^T) \mathbf{z}' \quad (6.27)$$

Note that the conventional Gaussian solution can be considered as a special case in our AR solution, since the distribution modeled by the AR(0) process is just the white Gaussian distribution. It can be verified that, for the AR(0) process, we have $\mathbf{T} = \mathbf{I}$, $\mathbf{M} = \sqrt{N} \boldsymbol{\Sigma}_w^{-1/2}$, $\mathbf{c} = \mathbf{0}$, and $\mathbf{z}' = N \mathbf{M} \bar{\mathbf{x}}$, and thus the conventional Gaussian solution for false data estimation given by Eq. (6.13) is equivalent to our AR solution given by Eq. (6.27).

Detection of Unobservable False Data Injection: The GLRT detector derived from Theorem 4 and Theorem 5 above has an underlying scenario that assumes the topological information (Jacobian matrix $\mathbf{H}$) is known to the operator/detector but unknown to the attacker, and detector's knowledge of both $\mathbf{H}$ and $\mathbf{X}$ gains an advantage against such attacks. Utilizing this advantage, it is

also able to detect certain unobservable attack schemes utilizing the knowledge of observation matrix $\mathbf{X}$.

In [165][148], the authors proposed an unobservable attack scheme that does not rely on the knowledge $\mathbf{H}$. Instead, the linear independent component analysis (ICA) based scheme only requires a number of sequential measurements at the steady-state, i.e., the observation matrix $\mathbf{X}$, to bypass traditional Gaussian detectors. The idea is to rewrite $\mathbf{H}\boldsymbol{\theta}$ as $\mathbf{H}\mathbf{A}\mathbf{y}$, where $\mathbf{A}$ is the unknown mixing matrix and $\mathbf{y}$ is the source vector of independent latent variables (components). Then, let $\mathbf{G} = \mathbf{H}\mathbf{A}$, we have $\mathbf{X} = \mathbf{H}\boldsymbol{\theta} + \mathbf{w} = \mathbf{G}\mathbf{y} + \mathbf{w}$. In a noise-free scenario, ICA infers both $\mathbf{G}$ and $\mathbf{y}$ so that $\mathbf{X} = \mathbf{G}\mathbf{y}$ with maximal independency/nongaussianity in $\mathbf{y}$. $\mathbf{G}$ has the same number of rows as $\mathbf{H}$, and its columns correspond to the estimated independent components in $\mathbf{y}$; the rows of $\mathbf{y}$ contain the independent components and the columns correspond to the N sequential observations. The inferred $\mathbf{y}$ is also called quasi-state vector.

In the context of state estimation, when the system dynamics changes within a small range, $|\mathbf{X} - \mathbf{G}\mathbf{y}|$ will be sufficiently smaller than a trivial number ϵ [165]. In this scenario, even without knowledge of the actual Jacobian matrix $\mathbf{H}$, the attacker can use the *virtual* Jacobian matrix $\mathbf{G}$ to generate false data with $\mathbf{G}\delta\mathbf{y}$, where $\delta\mathbf{y} = \theta_a$ is the false state in Theorem 4. Such attacks pose threat to the power system state estimation, and they provide an alternative way when the attackers could not gain access to the entire grid topology but a few snapshots of the system measurements are available.

Practically, the FastICA algorithm [166][167] is used to compute the two matrices $\mathbf{G}$ and $\mathbf{y}$. Given a threshold ε and an initial weight vector $\mathbf{w}$ of projection $\mathbf{y} = \mathbf{w}^T\mathbf{z}$, FastICA maximizes the nongaussianity of $\mathbf{y}$ by iteratively updating $\mathbf{w}$ and computing $\mathbf{G}$ from $\mathbf{w}^T\mathbf{G} = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The number of

independent components is initially set to the number of samples and iteratiely reduced if certain component has an eigenvalue smaller than a threshold.

Algorithm 5: The ICA based stealth false data scheme

Data: N sequential observations $\mathbf{X}$

Result: False data $\mathbf{X}'$

1. Compute $[\mathbf{G}, \mathbf{y}] = \text{FastICA}(\mathbf{X})$;

if $\max\{|\mathbf{X} - \mathbf{G}\mathbf{y}|\} < \epsilon$ **then**

 2. Generate $\delta\mathbf{y} \sim N(0, \sigma_y^2)$;

 3. Compute $\mathbf{X}' = \mathbf{X} + \mathbf{G}\delta\mathbf{y}$;

 4. Inject $\mathbf{X}'$ into the system as false measurements;

else

 break;

end

6.4 Simulations and Analysis

We conduct numerical simulations on the IEEE 30-bus power system to evaluate the detection performance for false data injection. We use MATPOWER, a Matlab package for power system simulation [168], to extract the measurement matrix $\mathbf{H}$ which has the size of 284×60 . There are 60 states in total which are the voltage amplitudes and angles on the 30 buses, and 284 meter measurements in all buses and branches. For the simplicity of our simulations, we simulate that the noises of all measurements in both observable and unobservable attacks have the same AR process: $e_{i,n} = 0.9 \times e_{i,n-1} + v_{i,n}$, where $v_{i,n} \sim (0, \sigma^2)$ for $i = 1, 2, \dots, M$ and $n = 1, 2, \dots, N$ ($M = 284$ and $N = 20$ are used in our simulations), and the constant false data with the magnitude of A is injected into D random meter measurements (i.e., $\|\mathbf{a}\|_0 = D$). All simulation results reported in this section are averaged over 10,000 independent runs.

For the observable attacks, we consider that 10% meter measurements (i.e., $D = 29$) can be manipulated by the attackers and that the magnitude of the constant false data is fixed at $A = 1$. Fig. 6.1 shows the receiver operating character-

Figure 6.1: Comparison of ROC curves in observable attacks when: (a) $\sigma^2 = 0.3$, $A = 1$, and $D = 29$; (b) $\sigma^2 = 0.5$, $A = 1$, and $D = 29$; (c) $\sigma^2 = 0.7$, $A = 1$, and $D = 29$

Figure 6.2: Comparison of ROC curves in ICA-based unobservable attacks when: (a) $\sigma^2 = 0.3$, $\sigma_y^2 = 0.3$, $A = 1$; (b) $\sigma^2 = 0.3$, $\sigma_y^2 = 0.5$, $A = 1$; (c) $\sigma^2 = 0.7$, $\sigma_y^2 = 0.5$, $A = 1$.

istic (ROC) curves of the AR detector and the Gaussian detector when $\sigma^2 = 0.3$, 0.5 and 0.7 . It can be shown that the performance of the Gaussian detector is degraded when the assumption of Gaussian noise is not satisfied. The superior performance of the AR detector further demonstrates the effectiveness of the proposed approaches.

For the ICA-based unobservable attacks, we consider all measurements are attackable and compare different combinations of σ^2 and σ_y^2 in three cases. Using the same setup for other parameters, the detection performances of unobservable false data attack are shown in Fig. 6.2. The AR detector shows competitive performance over the Gaussian detector against different σ_y^2 with a given σ^2 (Fig. 6.2a

and Fig. 6.2b) and remains robust against different σ^2 with a given σ_y^2 (Fig. 6.2b and Fig. 6.2c). The performance of Gaussian detector deteriorates significantly in unobservable attacks when the inference of virtual Jacobian matrix and quasi-state vector by the ICA based scheme.

6.5 Summary

This chapter investigates the security issue of smart grid. Particularly, it studies the problems of state estimation and false data injection detection in power systems, when the measurements are corrupted by the colored Gaussian noise. By modeling the colored Gaussian noise with the autoregressive process, we develop a state estimator and a false data detector to address these two problems. Numerical simulations were performed to demonstrate the effectiveness of the proposed methods. The superior performance of the proposed AR detector demonstrates the potential of AR detector against false data injection attacks in both observable and unobservable cases when the real Jacobian matrix remains secure and confidential to the attackers.

CHAPTER 7

Conclusions and Future Research Directions

7.1 Conclusions

This thesis provides a comprehensive study of intelligent cyber-physical systems in three aspects: machine learning algorithms, computing architectures, and applications. Next generation CPS require the implementation of advanced machine learning algorithms to incorporate intelligence into the physical processes and the adoption of high-performance computing architectures to support big data analysis and IoT. In this thesis, we contributed two novel advanced machine learning frameworks: EEF-based learning and ENN-based learning, which could be used in many aspects of learning including feature selection, distribution estimation, classification, anomaly detection, etc., and one hierarchical distributed Fog computing architecture which can offer high-performance computing over multiple spatial and temporal scales and support the domination of M2M communications in CPS. We also studied one important CPS application of smart grid, into which various learning methods are incorporated to enhance the safety of energy systems.

7.2 Original Contributions

The contributions of this thesis are three aspects in terms of machine learning algorithms, computing architectures and applications, in the context of various CPS. More specifically, the original contributions of this thesis are summarized as follows:

- A novel EEF-based learning framework has been developed for density estimation and classification. Class-wise PDFs can be constructed based on the reference distributions in an either data-driven manner or model-driven manner. The proposed method provides an effective way to address the problem

of model parameters learning using embedded latent variable exponential family probability. By incorporating the knowledge from reference distribution, it has been shown that superior learning performance can be achieved for many real-life applications.

- A novel ENN-based learning framework has been proposed for classification and anomaly detection. The proposed concept of ENN is capable to address the notable limitations of traditional KNN method. The “two-way communication” style make it be able to learn from the global distribution while keeping local information, therefore improving pattern recognition performance and providing a powerful technique for a wide range of data analysis applications in CPS. Using the concept of ENN, a RDOS method has been developed for anomaly detection which is one of the most important components to protect the safety of physical systems. The theoretical analysis has been provided to address the parameter setting issue which exists in many machine learning algorithms.
- For the learning in high-dimensional data, an optimal feature selection framework has been developed to select those most discriminative features for classification. Several efficient feature selection algorithms have also been proposed under this framework. To further address the information loss issue in feature selection, a generalized PDF projection theorem based on EEF has been developed to construct PDFs in the high-dimensional raw data space from the low-dimensional feature subspace. With the PDF construction, class-specific features can be used in the proposed classification method, which allows each class attains its own feature subset instead of a common feature subset for all classes as used in conventional approaches.

- A novel hierarchical distributed Fog computing architecture has been proposed to support high-performance computing and IoT. Fog computing, which extends the computing to the edge of network, is able to offer location-awareness and latency-sensitive monitoring and control. The hierarchical characteristic enables the domination of M2M communications through the network and the distributed characteristic offers the scalability of computing to support massive data analysis. Compared to the traditional high-performance computing architecture, the proposed Fog computing is more suitable for many CPS integrated with computations into massive number of infrastructure components. A working prototype is also constructed to experimentally evaluate the feasibility of the proposed architecture for smart pipeline monitoring.
- One important CPS application, smart grid, has been studied to enhance the security of power systems. Generalized state estimation and attack detection methods have been proposed and analyzed under different situations for the protection of energy systems against malicious cyber attacks. The simulation results and analysis have validated the effectiveness of the proposed approaches.

7.3 Future Research Directions

Three aspects of intelligent CPS: machine learning algorithms, computing architectures and applications, are studied in this thesis. Promising results, including experimental results, theoretical analysis and system prototypes, are provided to demonstrate the improved learning performance of the proposed machine learning algorithms, the feasibility of the proposed computing architecture, and the broad impact of investigated CPS applications. As CPS is an emerging research topic, there are still many opportunities to conduct further research along these three

aspects. For the machine learning algorithms, it is desirable to continue to explore the potentials of EEF-based and ENN-based learning approaches. For the computing architecture, it also would be seen that the proposed Fog computing could have a wide use in many CPS which require high-performance computing and M2M communication. In particular,

- An EEF-based transfer learning framework can be developed for facilitating the learning from a limited number of labeled data. Although the data is becoming cheap and abundant in many CPS as a large number of sensors are deployed, labeling the data is still very expensive because of high human labor cost. To address such limited supervision problem, it could be beneficial for the learning to incorporate the knowledge from related source domains. EEF provides a natural solution for knowledge incorporation, where the reference distribution can be learned from those source domains.
- Deep learning has brought many breakthroughs in machine learning with many more layers of neurons than ever before. With the improvement of learning criteria and the increasing number of data, deep learning is capable to learn hierarchical features from large-scale data sets and to produce remarkable learning performance improvement. Product of Experts (PoE) is the principle of Restricted Boltzmann Machine (RBM) which is the basis of deep learning. It is very interesting to generalize the PoE using EEF which allows to assign a weight for each expert, which would further increase the capacity of deep learning for many real-life problems in CPS.
- ENN has been proposed to address the imbalanced distribution issue existing in traditional nearest neighbor method, and hence an ENN-based imbalance learning method could be developed in the near future. For an imbalanced data set, it is necessary to measure the imbalance ratio among different

classes. The traditional measure, sample size ratio, may be inappropriate since it only reflects the imbalance of sample size. It is worth noticing that the class-wise generalized statistic in ENN can indicate the coherence of data for each class. Based on this generalized statistic, a more appropriate imbalance size ratio can be developed, and many existing imbalance learning schemes can be incorporated using the ENN-based measure.

- Many CPS need high-performance distributed computing to support decision making in spatial and temporal scales. The proposed Fog computing architecture is a perfect fit due to its scalability and distributed computing. It is a nature movement to apply this Fog computing architecture to many CPS, including smart healthcare, smart building, smart transportation, smart infrastructure, etc.

LIST OF REFERENCES

- [1] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.
- [2] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- [3] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.
- [4] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1, pp. 245–271, 1997.
- [5] G. H. John, R. Kohavi, K. Pfleger, *et al.*, "Irrelevant features and the subset selection problem," in *Machine learning: proceedings of the eleventh international conference*, 1994, pp. 121–129.
- [6] L. C. Molina, L. Belanche, and À. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in *IEEE International Conference on Data Mining*, 2002, pp. 306–313.
- [7] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- [8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.
- [9] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. NJ: Prentice-Hall: Englewood Cliffs, 1993.
- [10] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, Oct. 2007.
- [11] S. Kay, "Exponentially embedded families - new approach to model order estimation," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 333–345, 2005.
- [12] M. Rosenblatt *et al.*, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.
- [13] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, pp. 1065–1076, 1962.

- [14] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2009.
- [15] J. S. Simonoff, *Smoothing methods in statistics*. Springer Science & Business Media, 1996.
- [16] Q. Xie, C. A. Laszlo, and R. K. Ward, “Vector quantization technique for nonparametric classifier design,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, pp. 1326–1330, 1993.
- [17] R. Memisevic, L. Sigal, and D. J. Fleet, “Shared kernel information embedding for discriminative inference,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 778–790, 2012.
- [18] K. A. Patwardhan, G. Sapiro, and V. Morellas, “Robust foreground detection in video using pixel layers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 746–751, 2008.
- [19] X. Huang and D. N. Metaxas, “Metamorphs: deformable shape and appearance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1444–1459, 2008.
- [20] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
- [21] E. Fix and J. L. Hodges Jr, “Discriminatory analysis-nonparametric discrimination: consistency properties,” *U.S. Air Force Sch. Aviation Medicine*, Randolph Field, Texas, Project 21-49-004, Tech. Rep. 4, 1951.
- [22] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [23] P. Horton and K. Nakai, “Better prediction of protein cellular localization sites with the k nearest neighbors classifier,” in *Proc. of the 5th International Conference on Intelligent Systems for Molecular Biology*, vol. 5, 1997, pp. 147–152.
- [24] R. Parry, W. Jones, T. Stokes, J. Phan, R. Moffitt, H. Fang, L. Shi, A. Oberthuer, M. Fischer, W. Tong, and M. Wang, “ k -nearest neighbor models for microarray gene expression analysis and clinical outcome prediction,” *The Pharmacogenomics Journal*, vol. 10, no. 4, pp. 292–309, 2010.
- [25] J. H. Friedman, S. Steppel, and J. Tukey, *A nonparametric procedure for comparing multivariate point sets*. Stanford Linear Accelerator Center Computation Research Group Technical Memo, 1973, no. 153.

- [26] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the thirtieth Annual ACM Symposium on Theory of Computing*. ACM, 1998, pp. 604–613.
- [27] S. S. Craciunas, A. Haas, C. M. Kirsch, H. Payer, H. Röck, A. Rottmann, A. Sokolova, R. Trummer, J. Love, and R. Sengupta, "Information-acquisition-as-a-service for cyber-physical cloud computing," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 2010, pp. 14–14.
- [28] X. Xu, "From cloud computing to cloud manufacturing," *Robotics and computer-integrated manufacturing*, vol. 28, no. 1, pp. 75–86, 2012.
- [29] W. Zhu, C. Luo, J. Wang, and S. Li, "Multimedia cloud computing," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.
- [30] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.
- [31] H. Sandberg, A. Teixeira, and K. H. Johansson, "On security indices for state estimators in power networks," in *Workshop on Secure Control Systems*, Stockholm, Sweden, 2010.
- [32] G. Hug and J. A. Giampapa, "Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1362–1370, 2012.
- [33] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1244–1253, 2013.
- [34] R. Tan, V. B. Krishna, D. K. Yau, and Z. Kalbarczyk, "Integrity attacks on real-time pricing in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 2, p. 5, 2015.
- [35] Z.-H. Yu and W.-L. Chin, "Blind false data injection attack using pca approximation method in smart grid," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1219–1226, 2015.
- [36] S. Kullback, *Information Theory and Statistics*. Courier Dover Publication: 2nd ed., 1997.
- [37] T. Cover and J. Thomas, *Elements of Information Theory, 2nd Ed.* John Wiley and Sons, 2006.

- [38] H. Chernoff, “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” *The Annals of Mathematical Statistics*, pp. 493–507, 1952.
- [39] H. Jeffreys, “An invariant form for the prior probability in estimation problems,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.
- [40] J. Lin, “Divergence measures based on the shannon entropy,” *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.
- [41] K. Sawyer, “A multiple divergence criterion for testing between separate hypotheses,” *Statistics & Probability Letters*, vol. 1, no. 1, pp. 26–30, 1982.
- [42] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.
- [43] D. D. Lewis, “Naive (Bayes) at forty: The independence assumption in information retrieval,” in *European Conference on Machine Learning*, 1998, pp. 4–15.
- [44] A. McCallum and K. Nigam, “A comparison of event models for naive bayes text classification,” in *AAAI-98 Workshop on Learning for Text Categorization*, vol. 752, 1998, pp. 41–48.
- [45] V. N. Vapnik, *Estimation of dependences based on empirical data*. Springer-verlag New York, vol. 40.
- [46] S. Eyheramendy, D. D. Lewis, and D. Madigan, “On the Naive Bayes model for text categorization,” *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pp. 332–339, 2003.
- [47] D. Cai, X. He, and J. Han, “Document clustering using locality preserving indexing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.
- [48] D. Cai, Q. Mei, J. Han, and C. Zhai, “Modeling hidden topics on document manifold,” in *ACM Conference on Information and Knowledge Management*, 2008, pp. 911–920.
- [49] G. Skolidis and G. Sanguinetti, “Bayesian multitask classification with gaussian process priors,” *IEEE Trans. Neural Networks*, pp. 2011–2021, Dec. 2011.
- [50] M. Seeger, “Pac-bayesian generalisation error bounds for gaussian process classification,” *Machine Learning Research*, pp. 233–269, Oct. 2002.

- [51] C. M. Bishop, "Bayesian pca," *Advances in Neural Information Processing Systems*, pp. 382–388, 1999.
- [52] S. Kay, Q. Ding, and M. Rangaswamy, "Sensor intergration for classfication," *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2010.
- [53] M. Westover, "Asymptotic geometry of multiple hypothesis testing," *IEEE Trans. Inf. Theory*, pp. 3327–3329, Jul. 2008.
- [54] F. Nielsen and V. Garcia, "Statistical exponential families: A digest with flash cards," *CoRR*, vol. abs/0911.4863, 2009.
- [55] L. D. Brown, "Fundamentals of statistical exponential families: with application in statistical decision theory," *Institute of Mathematical Statistics, Hayworth, CA, USA*, 1986.
- [56] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [57] H. T. Kahraman, S. Sagiroglu, and I. Colak, "The development of intuitive knowledge classifier and the modeling of domain dependent data," *Knowledge-Based Systems*, vol. 37, no. 0, pp. 283 – 295, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705112002225>
- [58] K. J. Cios, D. K. Wedding, and N. Liu, "Clip3: cover learning using integer programming," *Kybernetes*, vol. 26, no. 5, pp. 513–536, 1997.
- [59] L. A. Kurgan, K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Gooden-day, "Knowledge discovery approach to automated cardiac spect diagnosis," *Artificial Intelligence in Medicine*, vol. 23, no. 2, pp. 149–169, 2001.
- [60] B. V. Ramana, M. S. P. Babu, and N. Venkateswarlu, "A critical study of selected classification algorithms for liver disease diagnosis," *International Journal of Database Management Systems*, vol. 3, no. 2, pp. 101–114, 2011.
- [61] S. Aeberhard, D. Coomans, and O. De Vel, "The classification performance of rda," *Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep.*, pp. 92–01, 1992.
- [62] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.
- [63] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," DTIC Document, Tech. Rep., 1985.

- [64] M. Samotyj, "The cost of power disturbance to industrial and digital economy companies," Consortium for electrical infrastructure to support a digital society, an initiative by EPRI and the Electrical Innovation Institute, Tech. Rep., Jun. 2000.
- [65] H. He and J. A. Starzyk, "A self organizing learning array system for power quality classification based on wavelet transformation," *IEEE Trans. Power Del.*, pp. 286–295, Jan. 2006.
- [66] T. K. A. Galil, M. Kamel, A. M. Youssef, E. F. E. Saadany, and M. M. A. Salama, "Power quality disturbance classification using the inductive interference approach," *IEEE Trans. Power Del.*, pp. 1812–1818, Oct. 2006.
- [67] H. Zhang, P. Liu, and O. P. Malik, "Detection and classification of power quality disturbances in noisy conditions," *Proc. Inst. Electr. Eng. Gener. Transm. Distrib.*, pp. 567–572, Sep. 2003.
- [68] H. T. Yang and C. C. Liao, "A de-noising scheme for enhancing wavelet-based power quality monitoring system," *IEEE Trans. Power Del.*, pp. 353–360, Jul. 2006.
- [69] P. O'Shea, "A high-resolution spectral analysis algorithm for power system disturbance monitoring," *IEEE Trans. Power Syst.*, pp. 676–680, Aug. 2002.
- [70] S. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. NJ: Prentice-Hall: Englewood Cliffs, 1998.
- [71] L. L. Scharf and B. Friedlander, "Matched subspace detector," *IEEE Trans. on Signal Processing*, pp. 2146–2157, Aug. 1994.
- [72] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.
- [73] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, no. 11, pp. 1771–1782, 2000.
- [74] W. H. Wong and L. Ma, "Optional pólya tree and bayesian inference," *The Annals of Statistics*, vol. 38, no. 3, pp. 1433–1459, 2010.
- [75] L. Lu, H. Jiang, and W. H. Wong, "Multivariate density estimation by bayesian sequential partitioning," *Journal of the American Statistical Association*, vol. 108, no. 504, pp. 1402–1410, 2013.
- [76] P. M. Baggenstoss, "Class-specific classifier: avoiding the curse of dimensionality," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 37–52, 2004.

- [77] B. Tang, H. He, P. Baggenstoss, and S. Kay, “A Bayesian classification approach using class-specific features for text categorization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1–1, 2016.
- [78] S. Kumar, J. Ghosh, and M. Crawford, “A hierarchical multiclassifier system for hyperspectral data analysis,” in *Multiple Classifier Systems*, 2000, pp. 270–279.
- [79] G. De Lannoy, D. François, and M. Verleysen, “Class-specific feature selection for one-against-all multiclass svms,” in *European Symposium on Artificial Neural Networks*, 2011, pp. 269–274.
- [80] B. Tang, H. He, Q. Ding, and S. Kay, “A parametric classification rule based on the exponentially embedded family,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 2, pp. 367–377, 2015.
- [81] Y. Yang and J. O. Pedersen, “A comparative study on feature selection in text categorization,” in *International Conference on Machine Learning*, vol. 97, 1997, pp. 412–420.
- [82] B. Tang, S. Kay, and H. He, “Toward optimal feature selection in naive bayes for text categorization,” *arXiv preprint arXiv:1602.02850*, 2016.
- [83] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [84] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in *European Conference on Machine Learning*. Springer, 1998.
- [85] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [86] M. F. Schilling, “Multivariate two-sample tests based on nearest neighbors,” *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 799–806, 1986.
- [87] M. Schilling, “Mutual and shared neighbor probabilities: Finite-and infinite-dimensional results,” *Advances in Applied Probability*, vol. 18, no. 2, pp. 388–405, 1986.
- [88] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

- [89] M. Lichman, *UCI Machine Learning Repository*, Web: <http://archive.ics.uci.edu/ml/>. School of Information and Computer Science, Irvine, CA: University of California, 2013.
- [90] J. H. Friedman, J. L. Bentley, and R. A. Finkel, “An algorithm for finding best matches in logarithmic expected time,” *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.
- [91] T. Liu, A. W. Moore, and A. Gray, “New algorithms for efficient high-dimensional nonparametric classification,” *The Journal of Machine Learning Research*, vol. 7, pp. 1135–1158, 2006.
- [92] S. Z. Li, K. L. Chan, and C. Wang, “Performance evaluation of the nearest feature line method in image classification and retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1335–1349, 2000.
- [93] K. C. Gowda and G. Krishna, “The condensed nearest neighbor rule using the concept of mutual nearest neighborhood,” *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 488–490, 1979.
- [94] W. Jin, A. K. Tung, and J. Han, “Mining top-n local outliers in large databases,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 293–298.
- [95] V. Hautamaki, I. Karkkainen, and P. Franti, “Outlier detection using k-nearest neighbour graph,” in *17th International Conference on Pattern Recognition*, 2004, pp. 430–433.
- [96] V. Barnett and T. Lewis, *Outliers in statistical data*. Wiley New York, 1994, vol. 3.
- [97] E. M. Knox and R. T. Ng, “Algorithms for mining distancebased outliers in large datasets,” in *Proceedings of the International Conference on Very Large Data Bases*, 1998, pp. 392–403.
- [98] C. C. Aggarwal and P. S. Yu, “Outlier detection for high dimensional data,” in *ACM Sigmod Record*, vol. 30, no. 2, 2001, pp. 37–46.
- [99] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” in *ACM SIGMOD Record*, vol. 29, no. 2, 2000, pp. 427–438.
- [100] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

- [101] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: A new data clustering algorithm and its applications,” *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.
- [102] M. Brito, E. Chavez, A. Quiroz, and J. Yukich, “Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection,” *Statistics & Probability Letters*, vol. 35, no. 1, pp. 33–42, 1997.
- [103] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93–104, 2000.
- [104] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, “Enhancing effectiveness of outlier detections for low density patterns,” in *Advances in Knowledge Discovery and Data Mining*, 2002, pp. 535–548.
- [105] K. Zhang, M. Hutter, and H. Jin, “A new local distance-based outlier detection approach for scattered real-world data,” in *Advances in Knowledge Discovery and Data Mining*, 2009, pp. 813–822.
- [106] W. Jin, A. K. Tung, J. Han, and W. Wang, “Ranking outliers using symmetric neighborhood relationship,” in *Advances in Knowledge Discovery and Data Mining*, 2006, pp. 577–593.
- [107] J. Tang, Z. Chen, A. W.-c. Fu, and D. Cheung, “A robust outlier detection scheme for large data sets,” in *In 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 2001.
- [108] L. J. Latecki, A. Lazarevic, and D. Pokrajac, “Outlier detection with kernel density functions,” in *Machine Learning and Data Mining in Pattern Recognition*, 2007, pp. 61–75.
- [109] J. Gao, W. Hu, Z. M. Zhang, X. Zhang, and O. Wu, “Rkof: robust kernel-based local outlier detection,” in *Advances in Knowledge Discovery and Data Mining*, 2011, pp. 270–283.
- [110] E. Schubert, A. Zimek, and H.-P. Kriegel, “Generalized outlier detection with flexible kernel density estimates,” in *Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA*, 2014, pp. 542–550.
- [111] V. A. Epanechnikov, “Non-parametric estimation of a multivariate probability density,” *Theory of Probability & Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.
- [112] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

- [113] S. G. Steckley, “Estimating the density of a conditional expectation,” Ph.D. dissertation, Cornell University, 2006.
- [114] M. Goldstein, “Unsupervised anomaly detection benchmark,” <http://dx.doi.org/10.7910/DVN/OPQMVF/>, 2015, online: Harvard Dataverse.
- [115] Nature, “Big data,” 2008, <http://www.nature.com/news/specials/bigdata/index.html>.
- [116] Z. Zhou, N. Chawla, Y. Jin, and G. Williams, “Big data opportunities and challenges: Discussions from data analytics perspectives,” *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–74, 2014.
- [117] L. Da Xu, W. He, and S. Li, “Internet of things in industries: a survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- [118] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, “An iot-oriented data storage framework in cloud computing platform,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.
- [119] K. Su, J. Li, and H. Fu, “Smart city and the applications,” in *International Conference on Electronics, Communications and Control (ICECC)*, 2011, pp. 1028–1031.
- [120] M. Castro, A. J. Jara, and A. F. Skarmeta, “Smart lighting solutions for smart cities,” in *International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2013, pp. 1374–1379.
- [121] S. Dey, A. Chakraborty, S. Naskar, and P. Misra, “Smart city surveillance: Leveraging benefits of cloud data stores,” in *IEEE Conference on Local Computer Networks Workshops*, pp. 868 – 876.
- [122] S. Yamamoto, S. Matsumoto, and M. Nakamura, “Using cloud technologies for large-scale house data in smart city,” in *IEEE 4th International Conference on Cloud Computing Technology and Science*, 2012, pp. 141–148.
- [123] N. Mitton, S. Papavassiliou, A. Puliafito, and K. S. Trivedi, “Combining cloud and sensors in a smart city environment,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–10, 2012.
- [124] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, “Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors,” in *International Conference on Sensor Technologies and Applications (SENSORCOMM)*, 2010, pp. 281–286.

- [125] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A hierarchical distributed Fog Computing architecture for big data analysis in smart cities," in *Proceedings of the ASE BigData & SocialInformatics 2015*, 2015, pp. 28:1–28:6.
- [126] F. Bonomi, R. Mito, P. Natarajan, and J. Zhu, "Fog computing: A platform for internet of things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.
- [127] J. Kane, B. Tang, Z. Chen, J. Yan, T. Wei, H. He, and Q. Yang, "Reflex-tree: A biologically inspired parallel architecture for future smart cities," in *International Conference on Parallel Processing (ICPP)*, 2015.
- [128] B. Anifowose, D. Lawler, D. Horst, and L. Chapman, "Evaluating interdiction of oil pipelines at river crossings using environmental impact assessments," *Area*, vol. 46, no. 1, pp. 4–17, 2014.
- [129] R. Alzbutas, T. Iešmantas, M. Povilaitis, and J. Vitkutė, "Risk and uncertainty analysis of gas pipeline failure and gas combustion consequence," *Stochastic Environmental Research and Risk Assessment*, vol. 28, no. 6, pp. 1431–1446, 2014.
- [130] M. Froggatt and J. Moore, "High-spatial-resolution distributed strain measurement in optical fiber with rayleigh scatter," *Applied Optics*, vol. 37, no. 10, pp. 1735–1740, 1998.
- [131] Z. Chen, G. Hefferman, and T. Wei, "Multiplexed oil level meter using a thin core fiber cladding mode exciter," *IEEE Photonics Technology Letters*, no. 99, pp. 1–1, 2015.
- [132] Z. Chen, L. Yuan, G. Hefferman, and T. Wei, "Ultraweak intrinsic fabry-perot cavity array for distributed sensing," *Optics letters*, vol. 40, no. 3, pp. 320–323, 2015.
- [133] Z. Chen, Y. Zeng, G. Hefferman, *et al.*, "Fiberid: Molecular-level secret for identification of things," in *Parallel Computing Technologies (PAR-COMPTech), 2015 National Conference on*, 2015, pp. 84–88.
- [134] Z. Chen, L. Yuan, G. Hefferman, and T. Wei, "Terahertz fiber bragg grating for distributed sensing," *IEEE Photonics Technology Letters*, vol. 27, no. 10, pp. 1084–1087, May 2015.
- [135] V. Sood, D. Fischer, J. Eklund, and T. Brown, "Developing a communication infrastructure for the smart grid," in *IEEE Electrical Power & Energy Conference (EPEC)*, 2009, pp. 1–7.
- [136] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.

- [137] S. Cui, Z. Han, S. Kar, T. Kim, H. Poor, and A. Tajer, “Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 106–115, 2012.
- [138] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.
- [139] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. S. Sastry, “Cyber security analysis of state estimators in electric power systems,” in *IEEE Conference on Decision and Control*, Atlanta, GA, 2010, pp. 5991–5998.
- [140] K. N. Plataniotis, D. Androutsos, and A. N. Venetsanopoulos, “Nonlinear filtering of non-gaussian noise,” *Journal of Intelligent and Robotic Systems*, vol. 19, no. 2, pp. 207–231, 1997.
- [141] I. Pitas and A. N. Venetsanopoulos, *Nonlinear digital filters: principles and applications*. Springer Science & Business Media, 2013, vol. 84.
- [142] W. Xu, M. Wang, and A. Tang, “On state estimation with bad data detection,” in *IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, 2011, pp. 5989–5994.
- [143] M. T. Hagh, S. M. Mahaei, and K. Zare, “Improving bad data detection in state estimation of power systems,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 1, no. 2, pp. 85–92, 2011.
- [144] A. Anwar and A. N. Mahmood, “Vulnerabilities of smart grid state estimation against false data injection attack,” in *Renewable Energy Integration*, 2014, pp. 411–428.
- [145] G. Dán and H. Sandberg, “Stealth attacks and protection schemes for state estimators in power systems,” in *IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, pp. 214–219.
- [146] L. Jia, R. Thomas, and L. Tong, “Malicious data attack on real-time electricity market,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5952–5955.
- [147] O. Vuković, K. C. Sou, G. Dán, and H. Sandberg, “Network-aware mitigation of data integrity attacks on power system state estimation,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1108–1118, 2012.
- [148] Y. Huang, M. Esmalifalak, H. Nguyen, R. Zheng, Z. Han, H. Li, and L. Song, “Bad data injection in smart grid: attack and defense mechanisms,” *IEEE Communications Magazine*, vol. 51, no. 1, pp. 27–33, 2013.

- [149] J. Kim and L. Tong, "On topology attack of a smart grid: Undetectable attacks and countermeasures," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1294–1305, 2013.
- [150] M. Esmalifalak, N. T. Nguyen, R. Zheng, and Z. Han, "Detecting stealthy false data injection using machine learning in smart grid," in *IEEE Conference on Global Communications*, Atlanta, GA, 2013, pp. 808–813.
- [151] F. Schweppe and J. Wildes, "Power system static-state estimation, part i: Exact model," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 120–125, 1970.
- [152] F. Schweppe and D. Rom, "Power system static-state estimation, part ii: Approximate model," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 125–130, 1970.
- [153] F. Schweppe, "Power system static-state estimation, part iii: Implementation," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 130–135, 1970.
- [154] A. Bose and K. Clements, "Real-time modeling of power networks," *IEEE Proceedings*, vol. 75, no. 12, pp. 1607–1622, 1987.
- [155] M. Filho, A. Leite da Silva, and D. Falcao, "Bibliography on power system state estimation (1968-1989)," *IEEE Transactions on Power Systems*, vol. 5, no. 3, pp. 950–961, 1990.
- [156] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC Press, 2004.
- [157] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. NJ: Prentice-Hall: Englewood Cliffs, 1993.
- [158] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.
- [159] A. Srivastava, T. Morris, T. Ernster, C. Vellaithurai, S. Pan, and U. Adhikari, "Modeling cyber-physical vulnerability of the smart grid with incomplete information," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 235–244, 2013.
- [160] Y. Yuan, Z. Li, and K. Ren, "Modeling load redistribution attacks in power systems," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 382–390, 2011.
- [161] W. Adkins and S. Weintraub, *Algebra: an approach via module theory*. Springer Science & Business Media, 2012, vol. 136.

- [162] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures,” in *IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, 2010, pp. 220–225.
- [163] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, “On malicious data attacks on power system state estimation,” in *45th International Universities Power Engineering Conference*, Cardiff, UK, 2010, pp. 1–6.
- [164] S. Kay, “Asymptotically optimal detection in unknown colored noise via autoregressive modeling,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 4, pp. 927–940, 1983.
- [165] M. Esmalifalak, H. Nguyen, R. Zheng, and Z. Han, “Stealth false data injection using independent component analysis in smart grid,” in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE, 2011, pp. 244–248.
- [166] A. Hyvärinen, “Fast and robust fixed-point algorithms for independent component analysis,” *Neural Networks, IEEE Transactions on*, vol. 10, no. 3, pp. 626–634, 1999.
- [167] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*. John Wiley & Sons, 2004, vol. 46.
- [168] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

BIBLIOGRAPHY

- Abur, A. and Exposito, A. G., *Power system state estimation: theory and implementation*. CRC Press, 2004.
- Adkins, W. and Weintraub, S., *Algebra: an approach via module theory*. Springer Science & Business Media, 2012, vol. 136.
- Aeberhard, S., Coomans, D., and De Vel, O., “The classification performance of rda,” *Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep.*, pp. 92–01, 1992.
- Aggarwal, C. C. and Yu, P. S., “Outlier detection for high dimensional data,” in *ACM Sigmod Record*, vol. 30, no. 2, 2001, pp. 37–46.
- Akaike, H., “A new look at the statistical model identification,” *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.
- Alzbutas, R., Iešmantas, T., Povilaitis, M., and Vitkutė, J., “Risk and uncertainty analysis of gas pipeline failure and gas combustion consequence,” *Stochastic Environmental Research and Risk Assessment*, vol. 28, no. 6, pp. 1431–1446, 2014.
- Anifowose, B., Lawler, D., Horst, D., and Chapman, L., “Evaluating interdiction of oil pipelines at river crossings using environmental impact assessments,” *Area*, vol. 46, no. 1, pp. 4–17, 2014.
- Anwar, A. and Mahmood, A. N., “Vulnerabilities of smart grid state estimation against false data injection attack,” in *Renewable Energy Integration*, 2014, pp. 411–428.
- Bache, K. and Lichman, M., “UCI machine learning repository,” 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>
- Baggenstoss, P. M., “Class-specific classifier: avoiding the curse of dimensionality,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 37–52, 2004.
- Barnett, V. and Lewis, T., *Outliers in statistical data*. Wiley New York, 1994, vol. 3.
- Bentley, J. L., “Multidimensional binary search trees used for associative searching,” *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- Bishop, C. M., *Pattern Recognition and Machine Learning*. Springer, Oct. 2007.

- Bishop, C. M., “Bayesian pca,” *Advances in Neural Information Processing Systems*, pp. 382–388, 1999.
- Blum, A. L. and Langley, P., “Selection of relevant features and examples in machine learning,” *Artificial Intelligence*, vol. 97, no. 1, pp. 245–271, 1997.
- Bonomi, F., Milito, R., Natarajan, P., and Zhu, J., “Fog computing: A platform for internet of things and analytics,” in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 169–186.
- Bonomi, F., Milito, R., Zhu, J., and Addepalli, S., “Fog computing and its role in the internet of things,” in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.
- Bose, A. and Clements, K., “Real-time modeling of power networks,” *IEEE Proceedings*, vol. 75, no. 12, pp. 1607–1622, 1987.
- Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J., “LOF: Identifying density-based local outliers,” *ACM SIGMOD Record*, vol. 29, no. 2, pp. 93–104, 2000.
- Brito, M., Chavez, E., Quiroz, A., and Yukich, J., “Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection,” *Statistics & Probability Letters*, vol. 35, no. 1, pp. 33–42, 1997.
- Brown, L. D., “Fundamentals of statistical exponential families: with application in statistical decision theory,” *Institute of Mathematical Statistics, Hayworth, CA, USA*, 1986.
- Cai, D., He, X., and Han, J., “Document clustering using locality preserving indexing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.
- Cai, D., Mei, Q., Han, J., and Zhai, C., “Modeling hidden topics on document manifold,” in *ACM Conference on Information and Knowledge Management*, 2008, pp. 911–920.
- Castro, M., Jara, A. J., and Skarmeta, A. F., “Smart lighting solutions for smart cities,” in *International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2013, pp. 1374–1379.
- Chen, Z., Heffernan, G., and Wei, T., “Multiplexed oil level meter using a thin core fiber cladding mode exciter,” *IEEE Photonics Technology Letters*, no. 99, pp. 1–1, 2015.
- Chen, Z., Yuan, L., Heffernan, G., and Wei, T., “Terahertz fiber bragg grating for distributed sensing,” *IEEE Photonics Technology Letters*, vol. 27, no. 10, pp. 1084–1087, May 2015.

- Chen, Z., Yuan, L., Hefferman, G., and Wei, T., “Ultraweak intrinsic fabry–perot cavity array for distributed sensing,” *Optics letters*, vol. 40, no. 3, pp. 320–323, 2015.
- Chen, Z., Zeng, Y., Hefferman, G., *et al.*, “Fiberid: Molecular-level secret for identification of things,” in *Parallel Computing Technologies (PARCOMPTECH), 2015 National Conference on*, 2015, pp. 84–88.
- Chernoff, H., “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations,” *The Annals of Mathematical Statistics*, pp. 493–507, 1952.
- Cios, K. J., Wedding, D. K., and Liu, N., “Clip3: cover learning using integer programming,” *Kybernetes*, vol. 26, no. 5, pp. 513–536, 1997.
- Comaniciu, D., Ramesh, V., and Meer, P., “Kernel-based object tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, 2003.
- Cortes, C. and Vapnik, V., “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- Cover, T. and Thomas, J., *Elements of Information Theory, 2nd Ed.* John Wiley and Sons, 2006.
- Cover, T. and Hart, P., “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- Craciunas, S. S., Haas, A., Kirsch, C. M., Payer, H., Röck, H., Rottmann, A., Sokolova, A., Trummer, R., Love, J., and Sengupta, R., “Information-acquisition-as-a-service for cyber-physical cloud computing,” in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 2010, pp. 14–14.
- Cui, S., Han, Z., Kar, S., Kim, T., Poor, H., and Tajer, A., “Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions,” *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 106–115, 2012.
- Da Xu, L., He, W., and Li, S., “Internet of things in industries: a survey,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.
- Dán, G. and Sandberg, H., “Stealth attacks and protection schemes for state estimators in power systems,” in *IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, pp. 214–219.

- De Lannoy, G., François, D., and Verleysen, M., “Class-specific feature selection for one-against-all multiclass svms,” in *European Symposium on Artificial Neural Networks*, 2011, pp. 269–274.
- Dey, S., Chakraborty, A., Naskar, S., and Misra, P., “Smart city surveillance: Leveraging benefits of cloud data stores,” in *IEEE Conference on Local Computer Networks Workshops*, pp. 868 – 876.
- Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern classification*. John Wiley & Sons, 2012.
- Epanechnikov, V. A., “Non-parametric estimation of a multivariate probability density,” *Theory of Probability & Its Applications*, vol. 14, no. 1, pp. 153–158, 1969.
- Esmalifalak, M., Nguyen, H., Zheng, R., and Han, Z., “Stealth false data injection using independent component analysis in smart grid,” in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE, 2011, pp. 244–248.
- Esmalifalak, M., Nguyen, N. T., Zheng, R., and Han, Z., “Detecting stealthy false data injection using machine learning in smart grid,” in *IEEE Conference on Global Communications*, Atlanta, GA, 2013, pp. 808–813.
- Ester, M., Kriegel, H.-P., Sander, J., and Xu, X., “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.
- Eyheramendy, S., Lewis, D. D., and Madigan, D., “On the Naive Bayes model for text categorization,” *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, pp. 332–339, 2003.
- Filho, M., Leite da Silva, A., and Falcao, D., “Bibliography on power system state estimation (1968-1989),” *IEEE Transactions on Power Systems*, vol. 5, no. 3, pp. 950–961, 1990.
- Filipponi, L., Vitaletti, A., Landi, G., Memeo, V., Laura, G., and Pucci, P., “Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors,” in *International Conference on Sensor Technologies and Applications (SENSORCOMM)*, 2010, pp. 281–286.
- Fix, E. and Hodges Jr, J. L., “Discriminatory analysis-nonparametric discrimination: consistency properties,” *U.S. Air Force Sch. Aviation Medicine*, Randolph Field, Texas, Project 21-49-004, Tech. Rep. 4, 1951.
- Friedman, J. H., Bentley, J. L., and Finkel, R. A., “An algorithm for finding best matches in logarithmic expected time,” *ACM Transactions on Mathematical Software*, vol. 3, no. 3, pp. 209–226, 1977.

- Friedman, J. H., Steppell, S., and Tukey, J., *A nonparametric procedure for comparing multivariate point sets*. Stanford Linear Accelerator Center Computation Research Group Technical Memo, 1973, no. 153.
- Froggatt, M. and Moore, J., “High-spatial-resolution distributed strain measurement in optical fiber with rayleigh scatter,” *Applied Optics*, vol. 37, no. 10, pp. 1735–1740, 1998.
- Galil, T. K. A., Kamel, M., Youssef, A. M., Saadany, E. F. E., and Salama, M. M. A., “Power quality disturbance classification using the inductive interference approach,” *IEEE Trans. Power Del.*, pp. 1812–1818, Oct. 2006.
- Gao, J., Hu, W., Zhang, Z. M., Zhang, X., and Wu, O., “Rkof: robust kernel-based local outlier detection,” in *Advances in Knowledge Discovery and Data Mining*, 2011, pp. 270–283.
- Giani, A., Bitar, E., Garcia, M., McQueen, M., Khargonekar, P., and Poolla, K., “Smart grid data integrity attacks,” *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1244–1253, 2013.
- Goldstein, M., “Unsupervised anomaly detection benchmark,” <http://dx.doi.org/10.7910/DVN/OPQMVF/>, 2015, online: Harvard Dataverse.
- Gowda, K. C. and Krishna, G., “The condensed nearest neighbor rule using the concept of mutual nearest neighborhood,” *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 488–490, 1979.
- Guyon, I. and Elisseeff, A., “An introduction to variable and feature selection,” *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
- Hagh, M. T., Mahaei, S. M., and Zare, K., “Improving bad data detection in state estimation of power systems,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 1, no. 2, pp. 85–92, 2011.
- Hautamaki, V., Karkkainen, I., and Franti, P., “Outlier detection using k-nearest neighbour graph,” in *17th International Conference on Pattern Recognition*, 2004, pp. 430–433.
- He, H. and Starzyk, J. A., “A self organizing learning array system for power quality classification based on wavelet transformation,” *IEEE Trans. Power Del.*, pp. 286–295, Jan. 2006.
- He, H. and Garcia, E. A., “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.
- Horton, P. and Nakai, K., “Better prediction of protein cellular localization sites with the k nearest neighbors classifier,” in *Proc. of the 5th International Conference on Intelligent Systems for Molecular Biology*, vol. 5, 1997, pp. 147–152.

- Huang, X. and Metaxas, D. N., “Metamorphs: deformable shape and appearance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1444–1459, 2008.
- Huang, Y., Esmalifalak, M., Nguyen, H., Zheng, R., Han, Z., Li, H., and Song, L., “Bad data injection in smart grid: attack and defense mechanisms,” *IEEE Communications Magazine*, vol. 51, no. 1, pp. 27–33, 2013.
- Hug, G. and Giampapa, J. A., “Vulnerability assessment of ac state estimation with respect to false data injection cyber-attacks,” *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1362–1370, 2012.
- Hyvärinen, A., “Fast and robust fixed-point algorithms for independent component analysis,” *Neural Networks, IEEE Transactions on*, vol. 10, no. 3, pp. 626–634, 1999.
- Hyvärinen, A., Karhunen, J., and Oja, E., *Independent component analysis*. John Wiley & Sons, 2004, vol. 46.
- Indyk, P. and Motwani, R., “Approximate nearest neighbors: Towards removing the curse of dimensionality,” in *Proceedings of the thirtieth Annual ACM Symposium on Theory of Computing*. ACM, 1998, pp. 604–613.
- Jeffreys, H., “An invariant form for the prior probability in estimation problems,” *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 186, no. 1007, pp. 453–461, 1946.
- Jia, L., Thomas, R., and Tong, L., “Malicious data attack on real-time electricity market,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011, pp. 5952–5955.
- Jiang, L., Da Xu, L., Cai, H., Jiang, Z., Bu, F., and Xu, B., “An iot-oriented data storage framework in cloud computing platform,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.
- Jin, W., Tung, A. K., and Han, J., “Mining top-n local outliers in large databases,” in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 293–298.
- Jin, W., Tung, A. K., Han, J., and Wang, W., “Ranking outliers using symmetric neighborhood relationship,” in *Advances in Knowledge Discovery and Data Mining*, 2006, pp. 577–593.
- Joachims, T., “Text categorization with support vector machines: Learning with many relevant features,” in *European Conference on Machine Learning*. Springer, 1998.

- John, G. H., Kohavi, R., Pfleger, K., *et al.*, “Irrelevant features and the subset selection problem,” in *Machine learning: proceedings of the eleventh international conference*, 1994, pp. 121–129.
- Kahraman, H. T., Sagiroglu, S., and Colak, I., “The development of intuitive knowledge classifier and the modeling of domain dependent data,” *Knowledge-Based Systems*, vol. 37, no. 0, pp. 283 – 295, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705112002225>
- Kane, J., Tang, B., Chen, Z., Yan, J., Wei, T., He, H., and Yang, Q., “Reflex-tree: A biologically inspired parallel architecture for future smart cities,” in *International Conference on Parallel Processing (ICPP)*, 2015.
- Kay, S., *Fundamentals of Statistical Signal Processing: Estimation Theory*. NJ: Prentice-Hall: Englewood Cliffs, 1993.
- Kay, S., *Fundamentals of Statistical Signal Processing: Estimation Theory*. NJ: Prentice-Hall: Englewood Cliffs, 1993.
- Kay, S., “Exponentially embedded families - new approach to model order estimation,” *IEEE Trans. Aerosp. Electron. Syst.*, pp. 333–345, 2005.
- Kay, S., Ding, Q., and Rangaswamy, M., “Sensor intergration for classficiation,” *Asilomar Conference on Signals, Systems, and Computers*, Nov. 2010.
- Kay, S., “Asymptotically optimal detection in unknown colored noise via autoregressive modeling,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 4, pp. 927–940, 1983.
- Kim, J. and Tong, L., “On topology attack of a smart grid: Undetectable attacks and countermeasures,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1294–1305, 2013.
- Knox, E. M. and Ng, R. T., “Algorithms for mining distancebased outliers in large datasets,” in *Proceedings of the International Conference on Very Large Data Bases*, 1998, pp. 392–403.
- Kohavi, R. and John, G. H., “Wrappers for feature subset selection,” *Artificial intelligence*, vol. 97, no. 1, pp. 273–324, 1997.
- Kosut, O., Jia, L., Thomas, R. J., and Tong, L., “Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures,” in *IEEE International Conference on Smart Grid Communications*, Gaithersburg, MD, 2010, pp. 220–225.
- Kosut, O., Jia, L., Thomas, R. J., and Tong, L., “On malicious data attacks on power system state estimation,” in *45th International Universities Power Engineering Conference*, Cardiff, UK, 2010, pp. 1–6.

- Kosut, O., Jia, L., Thomas, R. J., and Tong, L., "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645–658, 2011.
- Kullback, S., *Information Theory and Statistics*. Courier Dover Publication: 2nd ed., 1997.
- Kumar, S., Ghosh, J., and Crawford, M., "A hierarchical multiclassifier system for hyperspectral data analysis," in *Multiple Classifier Systems*, 2000, pp. 270–279.
- Kurgan, L. A., Cios, K. J., Tadeusiewicz, R., Ogiela, M., and Goodenday, L. S., "Knowledge discovery approach to automated cardiac spect diagnosis," *Artificial Intelligence in Medicine*, vol. 23, no. 2, pp. 149–169, 2001.
- Latecki, L. J., Lazarevic, A., and Pokrajac, D., "Outlier detection with kernel density functions," in *Machine Learning and Data Mining in Pattern Recognition*, 2007, pp. 61–75.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- Lewis, D. D., "Naive (Bayes) at forty: The independence assumption in information retrieval," in *European Conference on Machine Learning*, 1998, pp. 4–15.
- Li, S. Z., Chan, K. L., and Wang, C., "Performance evaluation of the nearest feature line method in image classification and retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1335–1349, 2000.
- Lichman, M., *UCI Machine Learning Repository*, Web: <http://archive.ics.uci.edu/ml/>. School of Information and Computer Science, Irvine, CA: University of California, 2013.
- Lin, J., "Divergence measures based on the shannon entropy," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 145–151, 1991.
- Liu, H. and Yu, L., "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.
- Liu, T., Moore, A. W., and Gray, A., "New algorithms for efficient high-dimensional nonparametric classification," *The Journal of Machine Learning Research*, vol. 7, pp. 1135–1158, 2006.

- Liu, Y., Ning, P., and Reiter, M. K., “False data injection attacks against state estimation in electric power grids,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.
- Lu, L., Jiang, H., and Wong, W. H., “Multivariate density estimation by bayesian sequential partitioning,” *Journal of the American Statistical Association*, vol. 108, no. 504, pp. 1402–1410, 2013.
- McCallum, A. and Nigam, K., “A comparison of event models for naive bayes text classification,” in *AAAI-98 Workshop on Learning for Text Categorization*, vol. 752, 1998, pp. 41–48.
- Memisevic, R., Sigal, L., and Fleet, D. J., “Shared kernel information embedding for discriminative inference,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 778–790, 2012.
- Mitton, N., Papavassiliou, S., Puliafito, A., and Trivedi, K. S., “Combining cloud and sensors in a smart city environment,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–10, 2012.
- Moghaddam, B., Jebara, T., and Pentland, A., “Bayesian face recognition,” *Pattern Recognition*, vol. 33, no. 11, pp. 1771–1782, 2000.
- Moghaddam, B. and Pentland, A., “Probabilistic visual learning for object representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696–710, 1997.
- Molina, L. C., Belanche, L., and Nebot, À., “Feature selection algorithms: A survey and experimental evaluation,” in *IEEE International Conference on Data Mining*, 2002, pp. 306–313.
- Monticelli, A., “Electric power system state estimation,” *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262–282, 2000.
- Nature, “Big data,” 2008, <http://www.nature.com/news/specials/bigdata/index.html>.
- Nielsen, F. and Garcia, V., “Statistical exponential families: A digest with flash cards,” *CoRR*, vol. abs/0911.4863, 2009.
- O’Shea, P., “A high-resolution spectral analysis algorithm for power system disturbance monitoring,” *IEEE Trans. Power Syst.*, pp. 676–680, Aug. 2002.
- Parry, R., Jones, W., Stokes, T., Phan, J., Moffitt, R., Fang, H., Shi, L., Oberthuer, A., Fischer, M., Tong, W., and Wang, M., “ k -nearest neighbor models for microarray gene expression analysis and clinical outcome prediction,” *The Pharmacogenomics Journal*, vol. 10, no. 4, pp. 292–309, 2010.

- Parzen, E., “On estimation of a probability density function and mode,” *The Annals of Mathematical Statistics*, pp. 1065–1076, 1962.
- Patwardhan, K. A., Sapiro, G., and Morellas, V., “Robust foreground detection in video using pixel layers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 746–751, 2008.
- Pitas, I. and Venetsanopoulos, A. N., *Nonlinear digital filters: principles and applications*. Springer Science & Business Media, 2013, vol. 84.
- Plataniotis, K. N., Androutsos, D., and Venetsanopoulos, A. N., “Nonlinear filtering of non-gaussian noise,” *Journal of Intelligent and Robotic Systems*, vol. 19, no. 2, pp. 207–231, 1997.
- Ramana, B. V., Babu, M. S. P., and Venkateswarlu, N., “A critical study of selected classification algorithms for liver disease diagnosis,” *International Journal of Database Management Systems*, vol. 3, no. 2, pp. 101–114, 2011.
- Ramaswamy, S., Rastogi, R., and Shim, K., “Efficient algorithms for mining outliers from large data sets,” in *ACM SIGMOD Record*, vol. 29, no. 2, 2000, pp. 427–438.
- Rifkin, R. and Klautau, A., “In defense of one-vs-all classification,” *The Journal of Machine Learning Research*, vol. 5, pp. 101–141, 2004.
- Rosenblatt, M. *et al.*, “Remarks on some nonparametric estimates of a density function,” *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J., “Learning internal representations by error propagation,” DTIC Document, Tech. Rep., 1985.
- Saeys, Y., Inza, I., and Larrañaga, P., “A review of feature selection techniques in bioinformatics,” *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
- Samotyj, M., “The cost of power disturbance to industrial and digital economy companies,” Consortium for electrical infrastructure to support a digital society, an initiative by EPRI and the Electrical Innovation Institute, Tech. Rep., Jun. 2000.
- Sandberg, H., Teixeira, A., and Johansson, K. H., “On security indices for state estimators in power networks,” in *Workshop on Secure Control Systems*, Stockholm, Sweden, 2010.
- Sawyer, K., “A multiple divergence criterion for testing between separate hypotheses,” *Statistics & Probability Letters*, vol. 1, no. 1, pp. 26–30, 1982.

- Scharf, L. L. and Friedlander, B., “Matched subspace detector,” *IEEE Trans. on Signal Processing*, pp. 2146–2157, Aug. 1994.
- Schilling, M. F., “Multivariate two-sample tests based on nearest neighbors,” *Journal of the American Statistical Association*, vol. 81, no. 395, pp. 799–806, 1986.
- Schilling, M., “Mutual and shared neighbor probabilities: Finite-and infinite-dimensional results,” *Advances in Applied Probability*, vol. 18, no. 2, pp. 388–405, 1986.
- Schubert, E., Zimek, A., and Kriegel, H.-P., “Generalized outlier detection with flexible kernel density estimates,” in *Proceedings of the 14th SIAM International Conference on Data Mining (SDM), Philadelphia, PA*, 2014, pp. 542–550.
- Schweppe, F., “Power system static-state estimation, part iii: Implementation,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 130–135, 1970.
- Schweppe, F. and Rom, D., “Power system static-state estimation, part ii: Approximate model,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 125–130, 1970.
- Schweppe, F. and Wildes, J., “Power system static-state estimation, part i: Exact model,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-89, no. 1, pp. 120–125, 1970.
- Scott, D. W., *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 2009.
- Seeger, M., “Pac-bayesian generalisation error bounds for gaussian process classification,” *Machine Learning Research*, pp. 233–269, Oct. 2002.
- Simonoff, J. S., *Smoothing methods in statistics*. Springer Science & Business Media, 1996.
- S.Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. NJ: Prentice-Hall: Englewood Cliffs, 1998.
- Skolidis, G. and Sanguinetti, G., “Bayesian multitask classification with gaussian process priors,” *IEEE Trans. Neural Networks*, pp. 2011–2021, Dec. 2011.
- Sood, V., Fischer, D., Eklund, J., and Brown, T., “Developing a communication infrastructure for the smart grid,” in *IEEE Electrical Power & Energy Conference (EPEC)*, 2009, pp. 1–7.

- Srivastava, A., Morris, T., Ernster, T., Vellaithurai, C., Pan, S., and Adhikari, U., “Modeling cyber-physical vulnerability of the smart grid with incomplete information,” *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 235–244, 2013.
- Steckley, S. G., “Estimating the density of a conditional expectation,” Ph.D. dissertation, Cornell University, 2006.
- Su, K., Li, J., and Fu, H., “Smart city and the applications,” in *International Conference on Electronics, Communications and Control (ICECC)*, 2011, pp. 1028–1031.
- Tan, R., Krishna, V. B., Yau, D. K., and Kalbarczyk, Z., “Integrity attacks on real-time pricing in electric power grids,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 2, p. 5, 2015.
- Tang, B., He, H., Baggenstoss, P., and Kay, S., “A Bayesian classification approach using class-specific features for text categorization,” *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, no. 99, pp. 1–1, 2016.
- Tang, B., Chen, Z., Hefferman, G., Wei, T., He, H., and Yang, Q., “A hierarchical distributed Fog Computing architecture for big data analysis in smart cities,” in *Proceedings of the ASE BigData & SocialInformatics 2015*, 2015, pp. 28:1–28:6.
- Tang, B., He, H., Ding, Q., and Kay, S., “A parametric classification rule based on the exponentially embedded family,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 2, pp. 367–377, 2015.
- Tang, B., Kay, S., and He, H., “Toward optimal feature selection in naive bayes for text categorization,” *arXiv preprint arXiv:1602.02850*, 2016.
- Tang, J., Chen, Z., Fu, A. W.-c., and Cheung, D., “A robust outlier detection scheme for large data sets,” in *In 6th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 2001.
- Tang, J., Chen, Z., Fu, A. W.-C., and Cheung, D. W., “Enhancing effectiveness of outlier detections for low density patterns,” in *Advances in Knowledge Discovery and Data Mining*, 2002, pp. 535–548.
- Teixeira, A., Amin, S., Sandberg, H., Johansson, K. H., and Sastry, S. S., “Cyber security analysis of state estimators in electric power systems,” in *IEEE Conference on Decision and Control*, Atlanta, GA, 2010, pp. 5991–5998.
- Vapnik, V. N., *Estimation of dependences based on empirical data*. Springer-verlag New York, vol. 40.

- Vuković, O., Sou, K. C., Dán, G., and Sandberg, H., “Network-aware mitigation of data integrity attacks on power system state estimation,” *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1108–1118, 2012.
- Westover, M., “Asymptotic geometry of multiple hypothesis testing,” *IEEE Trans. Inf. Theory*, pp. 3327–3329, Jul. 2008.
- Wong, W. H. and Ma, L., “Optional pólya tree and bayesian inference,” *The Annals of Statistics*, vol. 38, no. 3, pp. 1433–1459, 2010.
- Xie, Q., Laszlo, C. A., and Ward, R. K., “Vector quantization technique for non-parametric classifier design,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, pp. 1326–1330, 1993.
- Xu, W., Wang, M., and Tang, A., “On state estimation with bad data detection,” in *IEEE Conference on Decision and Control and European Control Conference*, Orlando, FL, 2011, pp. 5989–5994.
- Xu, X., “From cloud computing to cloud manufacturing,” *Robotics and computer-integrated manufacturing*, vol. 28, no. 1, pp. 75–86, 2012.
- Yamamoto, S., Matsumoto, S., and Nakamura, M., “Using cloud technologies for large-scale house data in smart city,” in *IEEE 4th International Conference on Cloud Computing Technology and Science*, 2012, pp. 141–148.
- Yang, H. T. and Liao, C. C., “A de-noising scheme for enhancing wavelet-based power quality monitoring system,” *IEEE Trans. Power Del.*, pp. 353–360, Jul. 2006.
- Yang, Y. and Pedersen, J. O., “A comparative study on feature selection in text categorization,” in *International Conference on Machine Learning*, vol. 97, 1997, pp. 412–420.
- Yu, Z.-H. and Chin, W.-L., “Blind false data injection attack using pca approximation method in smart grid,” *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1219–1226, 2015.
- Yuan, Y., Li, Z., and Ren, K., “Modeling load redistribution attacks in power systems,” *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 382–390, 2011.
- Zhang, H., Liu, P., and Malik, O. P., “Detection and classification of power quality disturbances in noisy conditions,” *Proc. Inst. Electr. Eng. Gener. Transm. Distrib.*, pp. 567–572, Sep. 2003.
- Zhang, K., Hutter, M., and Jin, H., “A new local distance-based outlier detection approach for scattered real-world data,” in *Advances in Knowledge Discovery and Data Mining*, 2009, pp. 813–822.

- Zhang, T., Ramakrishnan, R., and Livny, M., “BIRCH: A new data clustering algorithm and its applications,” *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.
- Zhou, Z., Chawla, N., Jin, Y., and Williams, G., “Big data opportunities and challenges: Discussions from data analytics perspectives,” *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–74, 2014.
- Zhu, W., Luo, C., Wang, J., and Li, S., “Multimedia cloud computing,” *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59–69, 2011.
- Zimmerman, R. D., Murillo-Sánchez, C. E., and Thomas, R. J., “MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.