DOCTORAL THESIS

Deep Learning Driven Data Analytics for Smart Grids

A thesis submitted for the degree of

Doctor of Philosophy

by

Fujia Han

Department of Electronic and Computer Engineering

Brunel University London

March 2019

Declaration of Authorship

I, Fujia Han, declare that the work in the dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific references in the text, the work is the candidate's own work. The work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

Signature: Fujia Han Date: 31/03/2019

Abstract

As advanced metering infrastructure (AMI) and wide area monitoring systems (WAMSs) are being deployed rapidly and widely, the conventional power grid is transitioning towards the smart grid at an increasing speed. A number of smart metering devices and real-time monitoring systems are capable to generate a huge volume of data on a daily basis. However, a variety of generated data can be made full use of to advance the development of the smart grid through big data analytics, especially, deep learning. Thus, the thesis is focused on data analysis for smart grids from three different aspects.

Firstly, a real-time data driven event detection method is presented, which is quite robust when dealing with corrupted and significantly noisy data of phase measurement units (PMUs). To be specific, the presented event detection method is based on a novel combination of random matrix theory (RMT) and Kalman filtering. Furthermore, a dynamic Kalman filtering technique is proposed through the adjustment of the measurement noise covariance matrix as the data conditioner of the presented method in order to condition PMU data. The experimental results show that the presented method is indeed quite robust in such practical situations that include significant levels of noisy or missing PMU data.

Secondly, a short-term residential load forecasting method is proposed on the basis of deep learning and k-means clustering, which is capable to extract similarity of residential load effectively and perform prediction accurately at the individual residential level. Specifically, it makes full use of k-means clustering to extract similarity among residential load and deep learning to extract complex patterns of residential load. In addition, in order to improve the forecasting accuracy, a comprehensive feature expression strategy is utilised to describe load characteristics of each time step in detail. The experimental results suggest that the proposed method can achieve a high forecasting accuracy in terms of both root mean square error (RMSE) and mean absolute error (MAE).

Thirdly, an online individual residential load forecasting method is developed based on

a combination of deep learning and dynamic mirror descent (DMD), which is able to predict residential load in real time and adjust the prediction error over time in order to improve the prediction performance. More specifically, it firstly employs a long short term memory (LSTM) network to build a prediction model offline, and then applies it online with DMD correcting the prediction error. In order to increase the prediction accuracy, a comprehensive feature expression strategy is used to describe load characteristics at each time step in detail. The experimental results indicate that the developed method can obtain a high prediction accuracy in terms of both RMSE and MAE.

To sum up, the proposed real-time event detection method contributes to the monitoring and operation of smart grids, while the proposed residential load forecasting methods contribute to the demand side response in smart grids.

Acknowledgement

Firstly, I would like to express my deepest gratitude to my first supervisor Prof. Maozhen Li for his invaluable guidance on my research and continuous support throughout my Ph.D. study. I would also like to express my thanks to my second supervisor Prof. Gareth Taylor for his financial support from the EU Horizon 2020 research project, TDX-ASSIST.

Also, I would like to thank my research colleagues for their constructive suggestions on my research, particularly, Dr. Barry Rawn and Dr. Phillip Ashton. And, I would like to thank my sincere friends at Brunel for their warm care and help over the past three years.

Last but not least, I would like to express my ultimate gratitude to my dear family for their support and encouragement during my Ph.D. study. In particular, I am genuinely grateful to my parents, who always give me considerate care and tremendous spiritual inspiration in my daily life. I would also like to be grateful to the one I have ever liked, who brings happiness and joy to my Ph.D. research life, and the one I will fall in love with in the future, who will make my whole life full of happiness and joy.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Advanced Metering Infrastructure	AMI
Artificial Neural Network	ANN
Convolutional Neural Network	CNN
Deep Belief Network	DBN
Deep Boltzmann Machine	DBM
Deep Feedforward Neural Network	DFNN
Deep Neural Network	DNN
Deep Policy Gradient	DPG
Deep Q-Learning	DQL
Deep Reinforcement Learning	DRL
Demand Response Program	DRP
Demand Side Management	DSM
Dynamic Mirror Descent	DMD
Energy Management System	EMS
Event Detector	ED
Exponential Linear Unit	ELU
Extreme Learning Machine	ELM
Gated Recurrent Unit	GRU
Generative Adversarial Network	GAN
Geographic Information System	GIS
Great Britain	GB
Hadoop Distributed File System	HDFS
High Performance Computing	HPC
Kalman Filter	KF
Least Absolute Shrinkage and Selection Operator	LASSO
Linear Eigenvalue Statistic	LES
Long Short Term Memory	LSTM
Mean Absolute Error	MAE
Mean Absolute Percentage Error	MAPE
Mean Spectral Radius	MSR
Non-Intrusive Load Monitoring	NILM

Phasor Measurement Unit	PMU
Pooling Deep Recurrent Neural Network	PDRNN
Principal Component Analysis	PCA
Random Matrix Model	RMM
Random Matrix Theory	RMT
Rectifier Linear Unit	ReLU
Recurrent Neural Network	RNN
Resilient Distributed Dataset	RDD
Root Mean Square Error	RMSE
Signal-to-Noise Ratio	SNR
Singular Value Decomposition	SVD
Stacked Denoising Autoencoder	SDAE
Supervisory Control and Data Acquisition	SCADA
Support Vector Machine	SVM
Support Vector Regression	SVR
Wide Area Monitoring System	WAMS

Chapter 1

Introduction

1.1 Background

An increasing number of efforts are being made across the world towards the promotion of clean sustainable energy for environmental protection and low carbon economy in order to meet its growing demand. The modern power system is being upgraded and transitioned to a more advanced power system to achieve this goal [1]. As the future development trend of the modern power network, the smart grid has great potentials to largely integrate renewable energy sources, reasonably control energy consumption patterns, and reliably ensure grid security. To be specific, the key benefits of the smart grid include uninterrupted power supply for all households, reduced transmission and distribution loss, high penetration of distributed renewable energy, large scale energy storage, flexibility for electricity customers to interact with electricity markets, market based electricity pricing, and demand side management (DSM) [2]. As a result, both developed and developing countries are paying great efforts to advance the development and implementation of the smart grid, such as Australia, Canada, Great Britain (GB), United States, Japan, Korea, and China.

1.1.1 Definitions of Smart Grid

In spite of this, there are a variety of definitions of Smart Grid all over the world, because different countries attempt to realise their own smart grid. Hence, there is no universal concept for Smart Grid at present [3]. The first official definition of Smart Grid was provided by the Energy Independence and Security Act of 2007, which was approved by the US Congress in January 2007. In addition, the concept of Smart Grid was developed in 2006 by the European Technology Platform for Smart Grids, and concerns an electricity network that can intelligently integrate the actions of all users connected to it, including generators, consumers, and those that do both, in order to efficiently deliver sustainable, economic, and secure electricity supplies. In China, State Grid Corporation of China interprets a smart grid as a strong robust electric power system, which is 1) supported by ultra-high voltage networks; 2) based on the coordinated development of power networks at different voltage levels; 3) aided by information and communication infrastructure; 4) characterised as an informationalised, automated, and interoperable power system; 5) a rational integration of electricity, information, and business flows.

Although there are various versions of definitions of Smart Grid in different countries, a common significant element for most definitions is the application of digital processing and communication to the power network, making data flow and information management central to the smart grid, as various capabilities mainly result from the deeply integrated use of digital technology with the power network [4].

1.1.2 Features of Smart Grids

Despite a wide range of diverse factors resulting in no agreement on a universal definition of Smart Grid, the main features of the smart grid can be summarised as follows:

1) Reliability

The smart grid makes full use of technologies, for example, state estimation, that improve event detection and allow self-healing of the grid without the intervention of electrical technicians. So, this will ensure more reliable power supply and reduce vulnerability to natural disasters and attacks to a large extent [5].

2) Flexibility in network topology

Smart grid infrastructure will be able to better handle possible bidirectional energy flows, allowing for distributed generation, such as photovoltaic panels, wind turbines, hydroelectric power, usage of fuel cells, and charging batteries of electric vehicles. Also, the smart grid aims to deal with those situations where a local sub-network generates more power than it is consuming and a reverse power flow appears [6].

3) Efficiency

Numerous contributions to overall efficiency improvement of energy infrastructure are expected from the application of smart grid technology, especially DSM, for instance, turning off air conditioners during short-term spikes in electricity price. The overall aim is less redundancy in transmission and distribution lines and greater utilization of generators, finally leading to lower electricity prices.

4) Load adjustment/balancing

The total load connected to the power network can probably change significantly over

time. Traditionally, load balancing strategies are designed to change customer consumption patterns to keep demand more uniform. However, in the smart grid, energy storage and distributed generation will enable load balancing without affecting customer behaviours.

5) Peak curtailment/levelling and time of use pricing

To motivate electricity users to reduce demand during the high cost peak usage period and perform peak curtailment or peak levelling, electricity prices are increased during the high demand period and decreased during the low demand period [7].

6) Sustainability

The enhanced flexibility of the smart grid enables greater penetration of highly variable renewable energy sources. Accordingly, smart grid technology is a necessary condition to realise a large scale integration of renewable electricity into the grid.

7) Market enabling

The smart grid allows for systematic communication between electricity suppliers and consumers, and enables both suppliers and consumers to be more flexible in their operational strategies. The overall effect is high energy efficiency and sensitive energy consumption patterns to time varying limitations of electricity supplies [8].

8) Demand response support

In the smart grid, demand response support permits generators and loads to interact in an automated way in real time, therefore coordinating demand to flatten spikes. This can also help users to cut their electricity bills by advising them to consume electricity when prices are lower [9].

With these main features, a further agreement on the definition of Smart Grid may be reached in the future, and new smart grid technologies can be developed on the basis of these characteristics.

1.1.3 Challenges and Issues of Smart Grids

Nowadays, the conventional power system is still transitioning towards the smart grid,

because smart grid technologies have not been advanced and mature enough to achieve this goal so far, which requires more academic and industrial researchers to focus on this research field. These main possible challenges and issues, which should be addressed in the future in order to completely implement the smart grid, can be outlined below [10]:

1) Self-healing action

As a crucial feature of the smart grid, self-healing is expected to recover the system and supply electricity to as many customers as possible, when a contingency happens. However, the major problem of self-healing action is uninterrupted power supply, which is closely involved with real-time monitoring of network operation, system state prediction, timely event detection, and rapid fault diagnosis [11-12].

2) Integration of renewable energy sources

Complex fluctuation and unpredictability of renewable energy sources, like wind power and photovoltaic power, require more sophisticated technologies to connect renewable generation to the power network in a more secure fashion. In addition, complex power electronic devices for integration bring harmonics to the power network, therefore affecting power quality negatively.

3) Energy storage

Although energy storage systems are capable to help to balance between supply and demand, their installation requires plenty of space and investment costs. On the other hand, due to diverse electricity consumption patterns of customers, energy storage devices are randomly used, some of which even remain unused for a long time, resulting in a waste of costs. Thus, effective energy management schemes are supposed to be developed in order to deal with this issue [13].

4) Customer participation

The smart grid motivates electricity consumers to actively participate in the energy management of the grid, and a wide range of demand response programs have been widely adopted to support consumers [14]. But, numerous consumers have not still raised enough awareness to get involved in these programs so far. Besides, during the communication between utilities and consumers, privacy invasion and data security may

be a major concern for both of them.

5) Reliability

High reliability is expected from the smart grid to ensure continuous power supply to electricity consumers. To meet this requirement, two potential challenges may be encountered. One is network automation, for instance, protection and control schemes, and the other one is network reconfiguration.

6) Interoperability and cyber security

The diversity of evolving smart grid technologies and applications is likely to bring significant challenges to network interoperability, as this interoperability involves a large number of monitoring and control activities, making possible bidirectional flows of electricity and information for generation, transmission, distribution, and consumption of various electric energy [15]. Meanwhile, information communication within the grid could probably be disrupted by such events as denial of service attacks, false data injection attacks, and cyber-physical switching attacks [16]. So, relevant information and communications technology (ICT) should be further researched and applied in the smart grid to enhance its cyber security.

With more efforts put on this series of potential challenges within the academic and industrial communities, the technological issues of the smart grid will be resolved gradually. Besides, the government should provide necessary political and technological support to boost the development of the smart grid.

1.2 Big Data Analytics

1.2.1 Overview of Big Data Analytics

During recent years, big data has increasingly obtained attention from both academia and industry all over the world. Due to its broad influence, the development of big data has profoundly changed our society and daily life and will continue to attract growing efforts from academic researchers and industrial experts in the future. As a well-known term in this information era, a uniform definition of big data has not still been achieved to date, because big data essentially means not only a huge volume of data, but also other unique characteristics that differentiate it from the concept of massive data.

However, International Data Corporation, a pioneer in researching big data and its impacts, gives a commonly recognised definition of big data. It describes big data technologies as new generation technologies and architectures aiming to economically extract valuable information from a huge amount of various data through enabling high-velocity capture, discovery, and/or analysis [17]. This definition indicates the four obvious features of big data, which are volume, variety, velocity, and value respectively. Accordingly, this 4V definition has been widely used to characterise big data nowadays.

In the meantime, with the rapid development of big data, big data analytics has gradually emerged and become increasingly popular in a majority of industries, such as health care, education, social media, finance, and information technology [18]. It is commonly believed that big data analytics can provide significant advantages to make timely and effective decisions relevant to time, cost, product development, service, optimization and so forth [19]. In general, big data analytics refers to applying analytical algorithms on powerful supportive platforms to uncover potentials within big data, for example, hidden patterns or unknown correlations. It normally consists of four consecutive phases, which are data generation, data acquisition, data storage, and data analysis respectively. Moreover, in terms of processing time, it can be classified into two categories. One is stream processing where data is analysed in real time as it arrives in a stream, and the other one is batch processing where data is firstly stored and then analysed later [17]. There are a few differences between these two processing methods, such as input, size, and storage. Despite its great advances, big data analytics are still faced with many challenges and issues, including algorithmic interpretation, modeling, prediction and simulation. These problems will have to be solved by the academic and industrial communities in order to bring more significant benefits to society and individuals in the future.

1.2.2 Contributions of Big Data Analytics to Smart Grids

Likewise, big data analytics has also been applied in the smart grid to develop a wide range of various applications over the past few years. Big data of the smart grid is generated from a series of different data sources, such as supervisory control and data acquisition (SCADA) systems, wide area monitoring systems (WAMSs), phasor measurement units (PMUs), smart meters, energy markets, weather, and geographic

information systems (GISs) [20-21]. According to these data sources, big data in the smart grid can be classified into two categories. One is internal data, such as WAMS data, PMU data, and energy management system (EMS) data, and the other one is external data, such as weather data, GIS data, and electricity user data [22]. As a result, based on vast amounts of data from the smart grid, the main contributions of big data analytics to the smart grid can be briefly summarised as the following aspects [23-28]:

1) Fault/Event detection

Real-time monitoring and control is crucial for reliability and operational efficiency of the smart grid, which is a strong driver for applications of big data analytics in the smart grid. The relevant applications include fault classification, fault location, anomaly alarming, and anomaly diagnosis. On the other hand, due to the rapid deployment of advanced monitoring devices across the power network, particularly PMUs, massive data is generated on a daily basis, therefore enabling these applications to significantly enhance reliability and operational security of the power network.

2) Integration of renewable energy

As renewable energy generation has great uncertainty and complex correlations with many factors, for example, wind speed of wind generation and solar irradiation of photovoltaic generation, big data analytics can be utilised to predict generation output in order to further increase the penetration of renewable energy generation. In addition, it is capable to conduct the simulation of renewable energy generation for power network planning and operation, such as average output and maximum output.

3) Integration of electric vehicles

Integrating electric vehicles into the smart grid plays a quite important role in environmental protection, owing to much less emission of pollution gas from them compared with traditional vehicles. However, integration of electric vehicles into the grid has still faced a great many problems so far, such as real-time dispatching of electric vehicles, charging pattern optimization, and battery consumption prediction. So, big data analytics can be applied to address these problems based on different kinds of data, e.g. battery information, journey information, and charging station information.

4) Demand side response

An increasing number of smart meters are being installed across the power network, therefore generating huge amounts of detailed electricity consumption data. As this massive smart meter data is capable to provide rich information on electricity consumption behaviours of consumers, it largely enhances the interaction between consumers and electric utilities and enables demand side response. More specifically, on the basis of smart meter data, big data analytics can be employed to develop demand response applications, e.g. short-term load forecasting, time-of-use pricing, and energy disaggregation (namely, non-intrusive load monitoring). Then, these demand response programs can provide electricity consumers with customized services and encourage them to actively participate in network operational management and finally reduce the costs of both consumers and utilities.

5) Smart grid cyber security

Advanced information and communications infrastructure has brought a wide range of great benefits to the smart grid. However, in the meantime, cyber security threats have extended to the smart grid and had severe impacts on the network operation. Hence, it is very important to realise cyber security situational awareness of the network. As network components, e.g. switches and routers, and security components, e.g. intrusion detection systems and access control systems, can generate a huge amount of security related data in the smart grid, then the cyber security issues of the network can be resolved by taking full advantage of big data analytics and security related data.

Obviously, the potential contributions of big data analytics to the smart grid are not only restricted to the ones discussed above. With the rapid advances of big data analytics, more big data analytical applications will be developed to boost the implementation of the smart grid.

1.3 Research Challenges and Objectives

Although many researchers have already made great efforts in the research field of applications of big data analytics in the smart grid, there are still a series of diverse challenges and issues to be resolved urgently. In the thesis, two aspects of research challenges are discussed in detail and then further solved through big data analytical techniques, in particular, deep learning.

1) Event detection

Situational awareness is a crucial function to provide an accurate understanding of power system dynamics in order that proper actions can be taken promptly to ensure secure and reliable operation of the power system. Event detection is regarded as very important part of situational awareness [29]. At present, event detection methods are mainly classified into model driven, e.g. neural network based approaches, decision tree based approaches, and principal component analysis based approaches, and data driven, e.g. wavelet based approaches, and rule based approaches. However, for model driven methods, it is generally accepted that if the training dataset is selected inappropriately or insufficiently, it will have a significantly negative effect on the performance of the method when it is applied online. Considering this fact, much more attention should be paid to the application of data driven methods in this research area. **As a result, the first objective of the thesis is to develop a data driven event detection method, which is particularly robust when applied to applications involving significantly polluted power system data.**

2) Residential load forecasting

Load forecasting has played a rather important role in the electrical power industry. Electrical power companies rely on short-term and long-term load forecasting at the feeder level to support network planning and operation, while electricity retailers make time-of-use pricing, procurement, and hedging decisions on the basis of load forecasting of electricity customers [27]. In reality, the load profile becomes much smoother as the load level increases. As a consequence, it is a quite challenging issue to forecast load at the residential level, due to high volatility of residential load.

A large number of research works on load forecasting at both medium voltage and high voltage levels have been presented up to now. For example, linear regression, least absolute shrinkage and selection operator (LASSO) regression, artificial neural networks (ANNs), and support vector machines (SVMs), have been explored and applied to perform load forecasting. It has already been verified that these methods are capable to achieve a satisfactory forecasting accuracy at medium voltage and high voltage levels. But, these traditional machine learning methods are not very suitable for residential load forecasting, because electricity consumption behaviours of household

residents are much more random. Considering this fact, some researchers have attempted to employ deep learning techniques to perform residential load forecasting, for instance, deep recurrent neural networks (RNNs). Although deep learning has great potential to extract more complex patterns of residential load, it is noted that it tends to incur overfitting, which has a seriously negative influence on its performance, if the training data is insufficient. **As a result, the second objective of the thesis is to develop a short-term residential load forecasting method, which is capable to avoid overfitting and achieve a high forecasting accuracy**.

In addition, numerous research works on residential load forecasting have been focused on offline applications. To be specific, existing residential load forecasting models are trained offline and applied online. However, if they are well trained offline and then applied online, they are very likely to have significantly poor performance, because residential load tends to change dramatically over time. Considering this fact, in order to support demand response programs, it is quite necessary to forecast residential load in real time (e.g. once every half an hour or once every a couple of hours) with online learning. **As a result, the third objective of the thesis is to develop an online residential load forecasting method, which is able to adjust the forecasting error in real time in order to improve the forecasting precision.**

1.4 Research Contributions

Based on the research challenges and objectives mentioned above, the main contributions of the thesis are detailed as follows:

1) A data driven event detection method based on random matrix theory and Kalman filtering

A real-time data driven event detection method is developed based on random matrix theory (RMT) and Kalman filtering, which is particularly robust when dealing with corrupted and significantly noisy samples of PMU data. Both simulated and real PMU data are utilised to verify its validity, and the experimental results show that it is very robust when applied in practical situations.

2) A dynamic Kalman filtering technique

A dynamic Kalman filtering technique is proposed through the adjustment of the measurement noise covariance matrix in order to condition PMU data. Simulated PMU data is used to testify its effectiveness, and the experimental results show that it is capable to perform noise reduction and data recovery effectively.

3) A short-term residential load forecasting method based on deep learning and k-means clustering

A short-term individual residential load forecasting method is presented on the basis of deep learning and k-means clustering, which is able to overcome overfitting and improve the forecasting accuracy. A real-life residential load dataset from Ireland is utilised to evaluate its performance, and the experimental results show that it can forecast residential load accurately.

4) An online residential load forecasting method based on deep learning and dynamic mirror descent

An online individual residential load forecasting method is designed based on deep learning and dynamic mirror descent (DMD), which is capable to adjust the forecasting error in real time. A real-life residential load dataset from Ireland is used to assess its performance, and the experimental results show that it can significantly improve the forecasting accuracy.

1.5 Thesis Layout

The remainder of the thesis is organized as follows:

Chapter 2 conducts a comprehensive literature review on applications of deep learning in power systems. To be specific, a brief overview of big data analytical techniques is firstly given from three major aspects, which are machine learning, deep learning, and high performance computing respectively. Then, a detailed review on deep learning applications in power systems is provided in terms of five major application fields, including load forecasting, demand response, renewable energy generation forecasting, event detection, and network operation and control.

Chapter 3 presents a data driven event detection method. It employs RMT as the

theoretical basis and mean spectral radius as an event indicator. Besides, as the data conditioner of the presented method, a dynamic Kalman filtering technique is developed in order to condition PMU data. The presented event detection method is tested on the standard IEEE 118-bus network and the GB transmission system to validate its effectiveness and robustness.

Chapter 4 proposes a short-term residential load forecasting method. It firstly makes use of k-means clustering to divide residential load into different groups according to load similarity, and then utilises deep learning to obtain a forecasting model for every group. The proposed residential load forecasting method is tested on an Irish residential load dataset from the Smart Metering Electricity Customer Behaviour Trials to evaluate its forecasting performance.

Chapter 5 designs an online residential load forecasting method. Different from the method proposed in Chapter 4, it is devoted to online residential load forecasting. More specifically, it firstly takes advantage of deep learning to build a well-trained forecasting model, and then applies the built model to forecast residential load with DMD adjusting the forecasting error in real time. The designed method is tested on an Irish residential load dataset to assess its ability to improve the forecasting precision.

Chapter 6 summarises the key findings of the research and the major contributions of the thesis, and provides a few potential research topics in the future work.

Chapter 2

Review of Applications of Deep Learning in Power Systems

2.1 Introduction

During the past few years, big data analytic techniques have been researched and developed at an increasing speed within both academia and industry. Due to a growing amount of various data from modern power networks, they have also played a vital role in the transition from modern power grids to future smart grids. As important part of big data analytic techniques, machine learning has already been widely applied in power systems up to now. However, as a huge amount of data is generated in modern power systems on a daily basis, deep learning is being adopted increasingly to develop a series of diverse big data applications for power systems because of its superior ability to extract extremely complex patterns of numerous data to that of machine learning.

So, with big data analytic techniques introduced briefly in the first place, this chapter aims to review common applications of deep learning in modern power systems, which are load forecasting, demand response, renewable energy generation forecasting, event detection, and network operation and control respectively.

2.2 Big Data Analytic Techniques

Admittedly, big data analytic techniques have been applied in a wide range of different industries and sectors in order to bring great benefits to society, business, and daily life in the big data era, as they are considered to be an extremely powerful tool to extract valuable information from big data. In general, big data analytic techniques consist mainly of three important components, namely, machine learning, deep learning, and high performance computing.

2.2.1 Machine Learning

Machine learning is a field of artificial intelligence that uses statistical techniques to give computer systems the ability to learn from data without being explicitly programmed. Within the area of data analytics, machine learning is a method used to devise complex models and algorithms, which allow researchers, analysts, engineers, and data scientists to produce reliable and repeatable decisions and results and uncover hidden insights through learning from historical relationships and trends within data.

In general, machine learning is typically classified into six categories, which are classification, regression, clustering, association analysis, density estimation, and dimensionality reduction respectively [30]. Classification and regression belong to supervised learning, while clustering, association analysis, density estimation, and dimensionality reduction belong to unsupervised learning. To be specific, classification algorithms consist mainly of k-nearest neighbors, decision trees, support vector machines, naive Bayes, and logistic regression; regression algorithms consist mainly of linear regression, ridge regression, LASSO regression, and regression trees; clustering algorithms involve k-means clustering and hierarchical clustering; association analysis algorithms involve Apriori and frequent pattern growth; density estimation algorithms include kernel density estimation and vector quantization; dimensionality reduction algorithms include principal component analysis (PCA) and singular value decomposition (SVD).

There have been a great many applications of machine learning in a wide range of research areas, such as image recognition, video processing, speech recognition, text processing, medical diagnosis, and renewable generation prediction. For example, reference [31] developed a new feature extraction method and combined it with machine learning methods to classify melanoma thickness. Reference [32] employed an SVM and a one-against-others decomposition method to perform texture classification without extra feature extraction. What is more, reference [33] presented a novel Bayesian classification approach to text classification, based on higher order dependencies between features and naive Bayes. Reference [34] proposed a smart board-level functional fault diagnosis method, which effectively combined SVMs and multiple kernel functions and incremental learning.

2.2.2 Deep Learning

Deep learning is part of a broader family of machine learning, which uses a cascade of multiple layers of nonlinear processing units for feature extraction and transformation and learns multiple levels of representations corresponding to different levels of abstraction [35]. Specifically, within deep learning, each current layer uses the output from the previous layer as its input and learns to transform its input into a slightly more abstract and composite representation. More importantly, a deep learning process can

learn which features to optically place at the most suitable level on its own.

Besides, deep learning models are vaguely inspired by information processing and communication patterns in biological nervous systems, but they have a variety of differences from the structural and functional properties of biological brains, therefore making them incompatible with neuroscience evidences. So far, a range of different deep learning structures have been developed and applied widely, such as deep Boltzmann machines (DBMs), deep belief networks (DBNs), deep feedforward neural networks (DFNNs), convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short term memory (LSTM) networks, and generative adversarial networks (GANs). Among all these structures, CNNs and RNNs (including LSTM networks) are the two most popular and common structures. It is generally accepted that CNNs are extremely suitable for dealing with spatial distribution data while RNNs have significant advantages in handling time series data [38-39, 46, 48, 55, 65].

Similar to machine learning, deep learning has also been applied to numerous research fields, including computer vision, automatic speech recognition, audio recognition, natural language processing, and machine translation, where they have produced desirable results comparable to and in some cases superior to human experts and machine learning because of its extraordinary learning capability. For instance, reference [36] proposed a novel acoustic modeling framework for speech recognition, combining multiple DFNNs with a hidden Markov model and a clustering technique. Reference [37] designed a sophisticated method to perform sentiment classification, based on a combination of DBNs and information geometry. In addition, reference [38] employed a CNN to conduct automatic segmentation of magnetic resonance images for quantitative analysis of brains. Reference [39] presented a new greedy layer-wise unsupervised pretraining strategy and used it to train a deep CNN to classify remote sensing images.

2.2.3 High Performance Computing

High performance computing (HPC) techniques normally refer to making use of multiple processors or a cluster of computers to process or analyse a huge amount of data. The main advantage of HPC techniques is that they are able to significantly speed

up the processing of numerous data with only a cluster of cost effective computers.

Among a few common HPC software frameworks, Hadoop and Spark are the two most popular software frameworks both in academia and industry at present, mainly because they are open source, freely available online, and well supported and maintained. To be specific, Hadoop has two core parts, which are the Hadoop distributed file system (HDFS) and MapReduce. HDFS provides users with the distributed file system for storing huge files easily, while MapReduce provides users with the distributed programming framework for fast computing. When a job is processed by Hadoop, MapReduce reads input data from disk, maps a function across the data, reduces the result of the map, and stores the reduction result on disk. However, different from Hadoop, Spark is a parallel computing framework aiming to process data and store intermediate results in distributed shared memory, as it has an essential distributed data framework, called the resilient distributed dataset (RDD), which makes Spark much faster than Hadoop in terms of data processing speed. RDDs have two types of computing operators, transformation and action. Transformation operators transform an RDD to another RDD, while action operators trigger Spark to submit jobs and output results. In addition, Spark has already included some helpful projects to enable users to develop applications conveniently, such as SparkSQL, Spark Streaming, GraphX, and MLlib.

Nowadays, a number of HPC applications have been developed to solve real-life problems. For examples, reference [40] introduced a new feature descriptor, namely, adaptive local motion descriptor, and combined it with a random forest to recognize human actions and deployed them in Spark. Reference [41] implemented a simple method in Hadoop to perform traffic demand prediction and designed an optimization module to improve the efficiency of Hadoop. Moreover, reference [42] proposed a practical massive video management platform using Hadoop, which is capable to process videos fast, e.g. video encoding and decoding, with desirable usability, performance, and availability.

2.3 Applications of Deep Learning in Power Systems

During the past few years, deep learning has been explored constantly in the electrical

power community to develop a variety of different applications, contributing substantially to the rapid advancement of power systems. The main applications of deep learning in power systems are reviewed in detail from five different aspects below [43].

2.3.1 Load/Demand Forecasting

Due to high penetration of distributed energy generation, scheduling and operation of modern power systems are faced with increasing challenges of uncertainty. As a result, accurate load forecasting at different levels becomes particularly important and helpful to address these challenges.

In reference [44], CNNs are combined with k-means clustering to forecast hourly electricity load. K-means clustering is firstly applied to group a large dataset containing more than one million of load records into subsets, and then these subsets are used to train CNNs. The experimental results show that the proposed approach is very effective. Reference [45] proposed a DBN embedded with parametric copula models to predict hourly load of a power network, using one year data from an urban area in the US. Both day ahead and week ahead predictions are performed, and the experimental results suggest that the proposed method is able to predict power load more accurately than classic artificial neural networks (ANNs), support vector regression (SVR), extreme learning machines (ELMs) in terms of mean absolute percentage error (MAPE) and root mean square error (RMSE). Likewise, reference [46] and reference [47] employed RNNs and DBNs respectively for short-term load forecasting of a power network. Different from reference [44-47], reference [48] presented a novel probabilistic forecasting approach combining LSTM networks and copula sampling, which is capable to predict intervals and densities of multiple variables, including power load, photovoltaic generation, wind generation, and electricity prices. This approach consists of two consecutive steps: firstly, determine the probabilistic forecasts for every variable during the whole forecasting period using LSTM networks; secondly, sample the resulting distribution of every variable using copula sampling. As a result, it takes both temporal information and cross-variable dependency into consideration so as to improve the prediction performance. In addition, reference [49] explored the application of deep neural networks (DNNs) with different activation functions and their combinations to load forecasting. Three activation functions, namely, sigmoid units, rectifier linear units

(ReLUs), and exponential linear units (ELUs), are selected and tested, and the preliminary results demonstrate that DNNs with ELUs outperform DNNs with others.

Overall, in existing research works, deep learning techniques are mainly applied together with other methods to obtain effective and accurate models for load forecasting. Besides, some external uncertain factors, for example, weather, could be considered and incorporated properly into forecasting models in order to further improve the forecasting precision. According to the references mentioned above, it can be also concluded that deep learning methods tend to outperform traditional machine learning methods, because they have a superior capability of dealing with large volumes of data and complex patterns. More relevant works on load forecasting will be introduced concretely in Chapter 4 and Chapter 5.

2.3.2 Demand Response

Identifying and forecasting energy flexibility on the demand side plays a crucial role in implementing demand response in power systems. The rapid and large-scale deployment of smart meters enables us to easily monitor energy consumption of electricity customers through real-time non-intrusive load monitoring (NILM). Reference [50] employed factored four way conditional restricted Boltzmann machines (FFW-CRBMs) and disjunctive FFW-CRBMs to conduct energy disaggregation on a real-life reference dataset. The experimental results suggest that disjunctive FFW-CRBMs perform slightly better than FFW-CRBMs for energy estimation, while both of them are almost comparable for energy identification. Reference [51] adopted three different DNN architectures for energy disaggregation, which are LSTM networks, denoising autoencoders, and vanilla DNNs respectively. A comprehensive comparison is made on a real-life aggregate power dataset from five domestic appliances between these three architectures and two baseline models using seven metrics. Similarly, reference [52] investigated the application of LSTM networks and gated recurrent unit (GRU) networks to NILM, and tested both on the UK domestic appliance level energy dataset using three different metrics.

Besides, deep learning techniques can be applied to perform classification of electricity customers, thus helping to analyse electricity consumption behaviours of customers. In

reference [53], RNNs are used to classify electricity users, and the experimental results prove that RNNs have better performance than some existing methods, such as k-nearest neighbors and hidden Markov models. In reference [54], a CNN is combined with an SVM to identify socio-demographic information of customers. The CNN firstly extracts features from load profiles and the SVM then classifies these extracted features. A comprehensive comparison is conducted on a real-life Irish smart meter dataset between this method and a few advanced machine learning methods.

Load prediction and control is also an important task of demand response, playing a crucial role in dynamic electricity pricing. Reference [55] utilised LSTM networks to forecast individual residential load, and the experimental results indicate that LSTM networks are very suitable for volatile residential load prediction. In addition, reference [56] explored deep Q-learning (DQL) and deep policy gradient (DPG) respectively for online building energy optimization, including building energy consumption and cost. The experimental results show that DPG is more suitable for online scheduling of energy resources than DQL, although both of them are able to successfully perform either minimization of energy cost or flattening of network energy profiles. Reference [57] proposed a novel approach to residential load control, based on deep learning and reinforcement learning. A CNN is used as a function approximator to estimate the Q-function within the fitted Q-iteration, and the experimental results validate its feasibility and effectiveness.

Demand response is a complex research topic for the smart grid, which is involved with consumption behaviours and social attributes of electricity customers, governmental policies, and so forth. Due to its extraordinary ability to extract hidden information, deep learning is regarded as a very promising technique to support demand response. But, it is still necessary to further explore effective combinations of deep learning and other techniques for energy flexibility analysis, load analysis and control, and decision making.

2.3.3 Renewable Energy Generation Forecasting

Renewable energy generation forecasting has a very important effect on secure integration of renewable energy sources into modern power networks and stable

network operation. Reference [58] used an LSTM model to predict wind power on the basis of numerical weather prediction data, and utilised PCA to extract the main input features fed to the LSTM model. The experimental results prove that the LSTM model has a higher prediction accuracy compared with the ANN and SVM models. Reference [59] presented a new multiscale wind power forecasting approach by establishing a multi-to-multi mapping network and an ensemble of stacked denoising autoencoders (SDAEs). Multiple numerical weather predictions are firstly corrected through an SDAE to generate better inputs, and a series of SDAEs and the generated inputs are then integrated into an ensemble of SDAEs to predict the individual wind power generated by different wind farms in a region. The experimental results of two real-world datasets testify the validity of the presented approach. Differing from reference [58-59], reference [60] investigated the application of DNNs to photovoltaic power forecasting, such as DBNs and LSTM networks. The experimental results show that the DNN models can achieve a higher prediction accuracy than a few reference models. Similarly, reference [61] and reference [62] applied LSTM networks and DBNs respectively to perform photovoltaic power generation forecasting.

Obviously, various deep learning methods have been increasingly developed and applied to renewable energy generation prediction, because deep learning has a number of superior characteristics, for example, high-level flexibility and excellent self-adaptive learning. However, it is expected that prediction precision can be further improved through combining deep learning with more data sources.

2.3.4 Event/Fault Detection

Event or fault detection, regardless of electrical equipment or networks, plays a very important role in ensuring secure power network operation. Nowadays, a large number of sensors and monitoring systems, such as SCADA systems and PMUs, are installed across the power network, enabling event or fault detection through taking full advantage of deep learning techniques.

Reference [63] designed an event detection method for power line insulators, which combines a CNN and an SVM to identify the power line insulator status. The CNN extracts features from aerial images of power line insulators, and then the SVM

classifies the extracted features. The experimental results on a real-world dataset of aerial images of power line insulators testify the validity of the designed method. Similarly, reference [64] applied a deep restricted Boltzmann machine jointly with an autoencoder to conduct fault detection for power transformers and circuit breakers. Both images and structured data with massive attributes are used as multiple inputs of the proposed model. The experimental results demonstrate that the proposed model outperforms some existing models in terms of accuracy.

Differing from reference [63-64], reference [65] introduced a novel deep learning based fault detection approach using power flow to identify topology changes in the power network. Power flow is firstly computed numerically, and then transformed to a computer-visualised image. After generating a number of power flow image samples of diverse system states and topologies, a CNN is trained and applied to perform fault detection. The experimental results suggest that the presented approach is capable to identify network faults effectively. Likewise, reference [66] and reference [67] employed SDAEs and LSTMs respectively to detect network events, and the experimental results prove their feasibility. Moreover, in reference [68], a new method to classify power quality events of the power network is presented based on deep learning techniques. Instead of voltage data, images of power quality event data are used as the input samples of the deep learning model. A real-world power quality event dataset from four different substations is utilised to test the presented method, and the experimental results validate its effectiveness. In reference [69], the potential of deep learning is explored in order to identify a series of power quality disturbances. For the purpose of understanding various deep learning architectures, a comprehensive comparison is made among different DNNs, such as CNNs, RNNs, LSTM networks, and GRU networks.

According to the references mentioned above, it can be easily summarised that both structured electrical data and images can be used as input samples by deep learning techniques to detect events or faults for electrical equipment or networks. But, there are still a number of problems to be resolved urgently, e.g. learning from a small dataset, selecting proper sample space, and identifying minor differences between normal and prefault conditions. More relevant works on event detection will be introduced in detail in Chapter 3.

2.3.5 Network Operation and Control

Frequency control, system security assessment, and stability analysis are very fundamental and important to ensure power network reliability. Therefore, WAMSs provide huge amounts of data to allow power network operators to have an accurate understanding of the network status via a wide range of deep learning applications based on WAMS data.

In reference [70], a load frequency control method is proposed to minimize frequency deviation, based on deep reinforcement learning (DRL) and continuous action search. The simulation results verify the superiority of the proposed method to a few existing methods. In reference [71], a novel deep learning based feature extraction framework is presented to assess power system security. Deep autoencoders are used to transform conventional state variables into a small number of abstract features, and an R-vine copula based model is proposed to sample historical data and generate massive system states for training. The superior performance of the presented framework is testified through a series of comparisons on a real-world dataset from the French transmission system. In addition, reference [72] designed a robust real-time load profile management framework for efficient power system operation. SDAEs are utilised to encode load profiles, and a locality sensitive hashing algorithm is implemented to cluster the encoded load profiles. The designed framework is tested on a real-world dataset of anonymous customers.

As for stability analysis, reference [73] proposed a DRL based load shedding scheme to improve short-term voltage stability, making use of two CNNs. One CNN is utilised to evaluate the quality of load shedding actions, and the other CNN is utilised to determine the time, location, and amount of load shedding. The simulation results on the modified IEEE 39-bus system validate the performance of the proposed scheme under different scenarios. Reference [74] presented a novel representational learning approach to transient stability assessment using PMU data. SDAEs firstly perform representational learning for crucial features, and a CNN is then added to the SDAEs to classify the system status. The simulation results on the IEEE 39-bus system demonstrate that the presented approach has a higher classification accuracy and greater robustness against

noisy PMU data compared to a few reference approaches. What is more, reference [75] combined a DBN with a simple linear classification model to identify network transient stability, while reference [76] employed a CNN to analyse network transient stability on the basis of the voltage phasor.

As modern power systems are becoming increasingly complex due to widespread integration of various new elements into power systems, e.g. wind generators, solar generators, flexible loads, and electric vehicles, deep learning is considered as a very promising alternative to conventional analysis methods for power network operation and control, because of its excellent ability to discover complicated patterns. On the other hand, as deep learning based methods are mostly regarded as model driven, their performance relies on sample selection and size to a large extent. As a result, it is generally suggested that combining model driven methods with data driven methods properly is very likely to improve analysis performance further.

Apparently, apart from the five common aspects detailed above, there are still some other uncommon applications of deep learning in power systems, including cyber security [77-80], electricity theft detection [81], dynamic energy management of microgrids [82], and energy trading behaviour modeling [83]. For example, reference [77] exploited deep learning techniques to identify the features of false data injection (FDI) attacks with historical measurement data and used the extracted features to detect FDI attacks in real time. The proposed detection mechanism is able to effectively relax the assumptions on potential attack scenarios and achieve a satisfactory accuracy. Besides, reference [81] presented a novel electricity theft detection approach based on a wide and deep CNN model, which consists of a wide component and a deep CNN component. The wide component captures the global features of one-dimensional electricity consumption data, while the deep CNN component recognises the nonperiodicity of electricity theft and the periodicity of normal electricity usage with two-dimensional electricity consumption data. The experimental results on a real-world dataset indicate that the presented approach outperforms other existing approaches.

2.4 Chapter Summary

This chapter firstly introduced big data analytic techniques briefly, in terms of machine

learning, deep learning, and high performance computing. Afterwards, it gave an overview of applications of deep learning in power systems from five main aspects, which are load forecasting, demand response, renewable energy generation forecasting, event detection, and network operation and control. At present, deep learning applications in power systems are mainly based on different effective combinations of deep learning techniques and other models in order to achieve desirable performance. However, there are still a series of challenges and issues to be addressed timely so that deep learning can be applied in more aspects of modern power systems.

Chapter 3

Robust Data Driven Event Detection Based on Random Matrix Theory and Kalman Filtering

3.1 Introduction

The development and increasing complexity of modern power systems is causing growing concerns over power system operational security, with network management and analysis becoming increasingly challenging. Traditional monitoring devices, employed in supervisory control and data acquisition (SCADA) systems, have already become quite impractical and inefficient for this challenge [84]. As a result, phasor measurement units (PMUs) are being deployed to measure phasors of bus voltages and currents, enabling real-time state monitoring in power systems [85-86]. Due to PMUs' higher sampling rates and accurate time synchronization, a huge amount of synchronous data is now being collated from power systems globally [87-88], making online monitoring possible through the detection of events.

At present, event detection methods are mainly classified into model driven (supervised) and data driven (unsupervised). For example, model driven methods involve neural network based approaches [89-90], decision tree based approaches [91-92], and principal component analysis based approaches [93-97]. However, for supervised learning, it is accepted that if the training dataset has been selected inappropriately or insufficiently, it will have a significantly negative effect on the performance of the method when it is applied online. Therefore, much attention has also been paid to the application of unsupervised learning in this field [98-101], including wavelet based approaches, energy function based approaches, and frequency difference based approaches.

Most recently, as a novel data driven approach, random matrix theory (RMT) was introduced and applied to early event detection in power systems [102]. In this case, as the statistical basis, random matrix models (RMMs) are first formed via the raw data from power systems. Then, several linear eigenvalue statistics (LESs) are designed through different test functions as indicators. Finally, by comparing the practical with theoretical values of LESs, event detection is conducted. However, in reality, as a major data source in modern power systems, PMU data is generally collected with significant noise levels and plenty of missing measurements, due to communication issues between PMUs and phasor data concentrators [103-104] and further issues in the measurement processes. When such an approach is faced with significantly noisy and missing data, it

will be unable to accurately detect abnormal events.

This chapter presents an event detection method which improves the performance of the RMT based method [102], making it much more robust when dealing with corrupted and significantly noisy samples of PMU data. To be specific, a real-time event detection method is developed based on a novel combination of RMT and Kalman filtering. Furthermore, a dynamic Kalman filtering technique is proposed through the adjustment of the measurement noise covariance matrix as the data conditioner of the presented method in order to condition PMU data. Finally, the presented event detection method is tested on the standard IEEE 118-bus network and the transmission system of Great Britain (GB) using the PMU data following actual transmission events in order to validate its feasibility and effectiveness. The experimental results show that the presented method is indeed much more robust in such practical situations that include significant levels of noisy or missing data.

3.2 Overview of Event Detection Methods

A large amount of research has been presented in the field of event detection for power systems. In reference [90], an approach to transmission line fault classification was presented on the basis of a combination of an adaptive resonance theory (ART) neural network and fuzzy logic. It applies a fuzzy decision rule to the output of the ART neural network to improve algorithm selectivity for a variety of real events. Reference [92] presented a new method, which is able to effectively classify traditional power system contingencies and cyber-attacks in real time. Specifically, it combines Hoeffding adaptive trees (HATs) with a drift detection method (DDM) and adaptive windowing (ADWIN), and involves a mechanism to forget old inferences and add new inferences to update the model continuously, because HATs are capable and suitable to process PMU data streams online and ADWIN and DDM can be used for change detection.

In addition, during recent years, principal component analysis (PCA) has become a popular and common approach to early event detection in power systems. For example, Xie analysed the dimensionality of PMU data under both normal and abnormal conditions, and employed PCA to detect abnormal events, based on the change of core subspaces of PMU data at the occurrence of an anomaly [93]. Meanwhile, this model

has an adaptive training mechanism, updating the model according to whether any event happens. Besides, linear dynamic system theory is introduced to prove the feasibility and validity of this algorithm. A similar work was carried out in reference [94], which uses a moving window PCA to detect and classify multiple events, including islanding, loss of load, and loss of generation. But, different from the one in reference [93] updating the PCA model on the basis of whether there is an event occurring, this method uses a moving window to update the PCA model in real time in order to adapt to the time varying behaviour of power systems. Moreover, it relies on the Hotelling's T^2 and Q statistics to determine if the power system is in an abnormal condition, while the method in reference [93] detects events based on the error between the predicted and measured values essentially. Another similar event detection method was developed by Guo in reference [95], only focusing on islanding detection for distributed generation systems. In contrast to the method in reference [94], Guo employed a recursive PCA algorithm to overcome the time varying behaviour of power systems for the purpose of reducing false alarms. Although a wide range of different machine learning techniques have been applied to event detection and classification, almost all the methods are supervised learning. In other words, most methods are based on models gained after training. But, as mentioned above, when the sample space is chosen improperly or insufficiently, an ill-trained model may be obtained.

In order to address this disadvantage of supervised learning, a few researchers have considered unsupervised learning techniques in this research area. For instance, Kim designed a wavelet based detection algorithm using PMU data [98]. The key idea of this algorithm is to monitor the energy of detail coefficients within the moving window in real time, and to determine whether the energy value exceeds the threshold or not. According to this idea, a modified wavelet energy function is defined to calculate the root mean square value of detail coefficients in the moving window, and the threshold is determined on the basis of the confidence level of event detection. Apart from this work, another data driven analytics method, which relies on the rules created by PMU data, was proposed for power system fault detection in reference [101]. It is comprised of the following three steps: 1) Identifying the fault bus; 2) Determining the fault type; 3) Detecting the fault line. Besides, fault thresholds are introduced in the rules and determined based on theoretical values and recorded PMU data during fault events.

However, it should be noted that the above unsupervised approaches need some thresholds to make decisions with respect to the occurrence of events. Admittedly, once thresholds are set up inaccurately, it will affect detection performance severely. Although the RMT based event detection approach was recently developed without setting any thresholds [102], it still did not take PMU data quality into consideration like the supervised and unsupervised approaches mentioned above. As a result, considering these drawbacks, this chapter proposes solutions to them.

The remainder of this chapter is organized as follows. Section 3.3 introduces random matrix theory. Section 3.4 formulates a dynamic quadratic prediction model based Kalman filtering technique. Section 3.5 details the combination of random matrix theory with Kalman filtering for event detection. Section 3.6 uses both simulated and real PMU data to evaluate the dynamic Kalman filtering technique and the proposed event detection method. Finally, the conclusions are presented in Section 3.7.

3.3 Random Matrix Theory

RMT has already proven to be effective for power system analysis, and a wide range of laws and theorems, such as Ring Law and Marchenko-Pastur Law, are included in RMT, which have been applied in power systems for different purposes [102, 105-106]. However, in this chapter, for the purpose of event detection, Ring Law, as the theoretical basis of data processing of event detection, and mean spectral radius (MSR), as the event indicator, will be briefly introduced, and then data processing of Ring Law will be formulated in detail.

3.3.1 Ring Law

Let $\mathbf{X}_{\mathrm{s}} \in \mathbf{C}^{N\times T}$ be a standard non-Hermitian random matrix, whose entries are independent and identically distributed (i.i.d.) variables with

$$\mu\left(\mathbf{x}_{\mathrm{s},i}\right) = 0,\ \sigma^2\left(\mathbf{x}_{\mathrm{s},i}\right) = 1 \quad (i = 1, 2, \cdots, N) \tag{3-1}$$

where $\mathbf{x}_{\mathrm{s},i}$ is the i^{th} row vector of $\mathbf{X}_{\mathrm{s}}$, μ denotes the mean, and σ^2 denotes the variance. For L standard non-Hermitian random matrices $\mathbf{X}_{\mathrm{s},i}$ $(i = 1, 2, \cdots, L)$, a matrix product is defined as

$$\mathbf{Z} = \prod_{i=1}^{L} \mathbf{X}_{\mathrm{u},i} \tag{3-2}$$

where $\mathbf{X}_{\mathrm{u},i}$ is the singular value equivalent of $\mathbf{X}_{\mathrm{s},i}$. The matrix product $\mathbf{Z}$ can be transformed to the standard matrix product $\mathbf{Z}_{\mathrm{s}}$, whose $\sigma^2(\mathbf{z}_{\mathrm{s},i}) = 1/N$ in each row. Thus, the empirical spectral density of $\mathbf{Z}_{\mathrm{s}}$ converges almost surely to the limit given by

$$f(\lambda_{\mathbf{Z}_{\mathrm{s}}}) = \begin{cases} \frac{1}{\pi c L} |\lambda_{\mathbf{Z}_{\mathrm{s}}}|^{\frac{2}{L}-2} & (1-c)^{\frac{L}{2}} \le |\lambda_{\mathbf{Z}_{\mathrm{s}}}| \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{3-3}$$

as $N, T \to \infty$ with a constant ratio $c = N/T \in (0, 1]$.

3.3.2 Mean Spectral Radius

Linear eigenvalue statistics indicate the statistical features of random matrices. An LES of a random matrix $\mathbf{X}$ is defined as

$$\mathrm{S}_n(\varphi) = \sum_{i=1}^{n} \varphi(\lambda_i) \tag{3-4}$$

where λ_i $(i = 1, 2, \cdots, n)$ are eigenvalues of $\mathbf{X}$, and $\varphi(\cdot)$ is a test function.

Mean spectral radius, as a special LES, is used to suggest the eigenvalue distribution of a random matrix. For the standard matrix product $\mathbf{Z}_{\mathrm{s}}$ (as mentioned above in Ring Law), MSR is formulated as

$$\mathrm{MSR} = \frac{1}{N} \sum_{i=1}^{N} |\lambda_{\mathbf{Z}_{\mathrm{s}},i}| \tag{3-5}$$

where $\lambda_{\mathbf{Z}_{\mathrm{s}},i}$ $(i = 1, 2, \cdots, N)$ are eigenvalues of $\mathbf{Z}_{\mathrm{s}}$, and $|\lambda_{\mathbf{Z}_{\mathrm{s}},i}|$ is the radius of $\lambda_{\mathbf{Z}_{\mathrm{s}},i}$ on the complex plane.

3.3.3 Data Processing of Ring Law

Within a raw data source $\mathbf{\Omega}$, a raw random matrix $\mathbf{X} \in \mathbf{C}^{N \times T}$ can be formed through a split window. Then, $\mathbf{X}$ can be transformed to a standard non-Hermitian matrix $\mathbf{X}_{\mathrm{s}} \in \mathbf{C}^{N \times T}$ row by row as follows:

$$\mathbf{x}_{\mathrm{s},i} = \left(\mathbf{x}_i - \mu(\mathbf{x}_i)\right) \frac{\sigma(\mathbf{x}_{\mathrm{s},i})}{\sigma(\mathbf{x}_i)} + \mu(\mathbf{x}_{\mathrm{s},i}) \quad (i = 1, 2, \cdots, N) \tag{3-6}$$

where $\mathbf{x}_i$ and $\mathbf{x}_{\mathrm{s},i}$ are the i^{th} row vectors of $\mathbf{X}$ and $\mathbf{X}_{\mathrm{s}}$ respectively, and $\mu(\mathbf{x}_{\mathrm{s},i}) = 0$, $\sigma^2(\mathbf{x}_{\mathrm{s},i}) = 1$.

Afterwards, the matrix $\mathbf{X}_{\mathrm{u}} \in \mathbf{C}^{N\times N}$ is introduced as the singular value equivalent of $\mathbf{X}_{\mathrm{s}}$ by

$$\mathbf{X}_{\mathrm{u}} = \sqrt{\mathbf{X}_{\mathrm{s}}\mathbf{X}_{\mathrm{s}}^{\mathrm{T}}}\mathbf{U} \tag{3-7}$$

where $\mathbf{X}_{\mathrm{s}}^{\mathrm{T}}$ is the transpose of $\mathbf{X}_{\mathrm{s}}$ and $\mathbf{U} \in \mathbf{C}^{N\times N}$ is a Haar unitary matrix, $\mathbf{X}_{\mathrm{u}}\mathbf{X}_{\mathrm{u}}^{\mathrm{T}} \equiv \mathbf{X}_{\mathrm{s}}\mathbf{X}_{\mathrm{s}}^{\mathrm{T}}$.

However, for a series of arbitrary non-Hermitian random matrices $\mathbf{X}_i$ $(i = 1, 2, \cdots, L)$ from the raw data source $\boldsymbol{\Omega}$, the matrix product $\mathbf{Z} = \prod_{i=1}^{L} \mathbf{X}_{\mathrm{u},i} \in \mathbf{C}^{N\times N}$ is obtained. Then, it is converted to the standard matrix product $\mathbf{Z}_{\mathrm{s}} \in \mathbf{C}^{N\times N}$ through the following formula:

$$\mathbf{z}_{\mathrm{s},i} = \frac{\mathbf{z}_i}{\sqrt{N}\sigma(\mathbf{z}_i)} \tag{3-8}$$

where $\mathbf{z}_i$ and $\mathbf{z}_{\mathrm{s},i}$ are the i^{th} row vectors of $\mathbf{Z}$ and $\mathbf{Z}_{\mathrm{s}}$ respectively.

3.4 Dynamic Kalman Filtering

As a state estimation technique, Kalman filtering has been one of the most common methods for conditioning PMU data in power systems. Many researchers have presented a wide range of works on applications of Kalman filtering to PMU data conditioning [107-109]. Most recently, a quadratic prediction model based Kalman filtering technique was proposed [110]. It is able to cleanse and recover PMU data easily with the quadratic prediction model reflecting the quadratic relationship between the past, present and future states. However, within this method, the measurement noise covariance matrix $\mathbf{R}$ is constant, and thus unable to condition PMU data accurately in some situations. As a result, in this chapter, this technique is improved by making the measurement noise covariance matrix $\mathbf{R}$ dynamic to adapt to the conditioning process.

The classic model of Kalman filtering is shown as follows:

$$\begin{cases} \mathbf{x}(t+1) = \boldsymbol{\Phi}(t+1, t)\mathbf{x}(t) + \boldsymbol{\Gamma}(t+1, t)\mathbf{w}(t) \\ \mathbf{z}(t+1) = \mathbf{H}(t+1)\mathbf{x}(t+1) + \mathbf{v}(t+1) \end{cases} \tag{3-9}$$

where $\mathbf{x}(t)$ and $\mathbf{x}(t+1)$ are the system states at time t and time t+1 respectively, $\mathbf{z}(t+1)$ is the actual measurement at time t+1, $\mathbf{w}(t)$ is the zero-mean Gaussian process noise at time t, $\mathbf{v}(t+1)$ is the zero-mean Gaussian measurement noise at time t+1,

$\boldsymbol{\Phi}(t+1,t)$ is the state transition matrix from time t to time t+1, $\boldsymbol{\Gamma}(t+1,t)$ is the disturbance transition matrix from time t to time t+1, and $\mathbf{H}(t+1)$ is the measurement matrix at time t+1.

For the purpose of optimization, the classic model of Kalman filtering can be expressed in the recursive way as

$$\begin{cases} \hat{\mathbf{x}}(t+1|t+1) = \boldsymbol{\Phi}(t+1,t)\hat{\mathbf{x}}(t|t) + \mathbf{K}(t+1)\big(\mathbf{z}(t+1) - \hat{\mathbf{z}}(t+1|t)\big) \\ \hat{\mathbf{z}}(t+1|t) = \mathbf{H}(t+1)\hat{\mathbf{x}}(t+1|t) \end{cases} \tag{3-10}$$

where $\mathbf{K}(t+1)$ is the Kalman gain matrix at time t+1, $\hat{\mathbf{z}}(t+1|t)$ is the a priori measurement of time t+1 given time t, $\hat{\mathbf{x}}(t+1|t)$ is the a priori estimate of time t+1 given time t, and $\hat{\mathbf{x}}(t|t)$ is the a posteriori estimate of time t given time t, which is also regarded as the final optimal estimate of the system state at time t.

According to reference [111], the quadratic prediction model is formulated by

$$\hat{x}(t+1|t) = 3\hat{x}(t|t) - 3\hat{x}(t-1|t-1) + \hat{x}(t-2|t-2) \tag{3-11}$$

it can be known that two adjacent state vectors share two of the three state variables. Specifically, each state vector is formed by a moving window, which contains three successive snapshots of the system states and moves forwards only one snapshot at a time generating the next state vector. Thus, in order to estimate the next state, the state vectors $\hat{\mathbf{x}}(t|t)$ and $\hat{\mathbf{x}}(t+1|t)$ are respectively expressed as follows [110]:

$$\hat{\mathbf{x}}(t|t) = \begin{bmatrix} \hat{x}(t|t) \\ \hat{x}(t-1|t-1) \\ \hat{x}(t-2|t-2) \end{bmatrix} \tag{3-12}$$

$$\hat{\mathbf{x}}(t+1|t) = \begin{bmatrix} \hat{x}(t+1|t) \\ \hat{x}(t|t) \\ \hat{x}(t-1|t-1) \end{bmatrix} \tag{3-13}$$

where $\hat{x}(t+1|t)$, $\hat{x}(t|t)$, $\hat{x}(t-1|t-1)$, and $\hat{x}(t-2|t-2)$ are all complex values, including magnitude and phase angle.

As for $\boldsymbol{\Phi}(t+1,t)$ and $\mathbf{H}(t+1)$, they can be expressed as constant matrices shown in (3-14) and (3-15) [110]:

$$\mathbf{\Phi}(t+1,t)=\begin{bmatrix}3 & -3 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix} \tag{3-14}$$

$$\mathbf{H}(t+1)=[1 \quad 0 \quad 0] \tag{3-15}$$

The Kalman gain matrix $\mathbf{K}(t+1)$ is specified by the following formulas [110]:

$$\begin{cases}\mathbf{K}(t+1)=\mathbf{P}(t+1|t)\mathbf{H}^{\mathrm{T}}(t+1)\times\left[\mathbf{H}(t+1)\mathbf{P}(t+1|t)\mathbf{H}^{\mathrm{T}}(t+1)+\mathbf{R}(t+1)\right]^{-1}\\ \mathbf{P}(t+1|t)=\mathbf{\Phi}(t+1,t)\mathbf{P}(t|t)\mathbf{\Phi}^{\mathrm{T}}(t+1,t)+\mathbf{\Gamma}(t+1,t)\mathbf{Q}(t)\mathbf{\Gamma}^{\mathrm{T}}(t+1,t)\\ \mathbf{P}(t+1|t+1)=[\mathbf{I}-\mathbf{K}(t+1)\mathbf{H}(t+1)]\mathbf{P}(t+1|t)\end{cases} \tag{3-16}$$

where $\mathbf{P}(t+1|t)$ is the a priori error covariance matrix of time t+1 given time t, $\mathbf{P}(t|t)$ is the a posteriori error covariance matrix of time t given time t, $\mathbf{I}$ is the identity matrix, $\mathbf{H}^{\mathrm{T}}$, $\mathbf{\Phi}^{\mathrm{T}}$, and $\mathbf{\Gamma}^{\mathrm{T}}$ are the transposes of $\mathbf{H}$, $\mathbf{\Phi}$, and $\mathbf{\Gamma}$ respectively, and $\mathbf{R}$ and $\mathbf{Q}$ are the measurement noise covariance matrix and process noise covariance matrix respectively, both of which are scalar values in this method. Furthermore, it is assumed that the Kalman filtering process starts at t=3 and $\hat{\mathbf{x}}(3|3)$ and $\mathbf{P}(3|3)$ are the matrix of ones and the matrix of zeros respectively. The purpose for this practice is to make the Kalman filter track the optimal estimate fast within a short period.

As the measurement noise covariance matrix $\mathbf{R}$ has a significant influence on the performance of Kalman filtering, this parameter will be adjusted dynamically in this chapter. First of all, a residue between the a posteriori estimate and the actual measurement is calculated at each time t as follows [112]:

$$r(t)=\hat{x}(t|t)-z(t) \tag{3-17}$$

where $r(t)$ is the residue at time t, and $z(t)$=$\mathbf{z}(t)$. For a period, a residue vector is formed as shown in (3-18):

$$\mathbf{r}(t+1)=[r(t-W+1) \quad r(t-W+2) \quad \cdots \quad r(t-1) \quad r(t)] \tag{3-18}$$

where $\mathbf{r}(t+1)$ is the residue vector at time t+1, and W is the time length. Then, the variance of the residue vector is calculated as the measurement noise covariance matrix shown below:

$$\mathbf{R}(t+1)=\mathrm{Var}\left(\mathbf{r}(t+1)\right) \tag{3-19}$$

where $\mathbf{R}(t+1)$ is the measurement noise covariance matrix at time t+1. Here, it is

noted that $\mathrm{Var}\left(\mathbf{r}(t+1)\right)$ is the variance of a series of complex values. According to (3-18) and (3-19), it can be easily seen that $\mathbf{R}(t+1)$ is obtained on the basis of W previous residues, namely, from time t-W+1 to time t.

Instead of keeping the measurement noise covariance matrix **R** constant, the dynamic Kalman filtering technique adjusts it continuously to reduce noise more effectively. To be specific, when the noise is heavier, the difference between the a posteriori estimate $\hat{x}(t|t)$ and the actual measurement $z(t)$ becomes greater, therefore causing the residue $r(t)$ to become greater, which finally leads to a greater measurement noise covariance matrix $\mathbf{R}(t+1)$. In this situation, as can be inferred from (3-10) and (3-16), to calculate the next a posteriori estimate $\hat{x}(t+1|t+1)$, the actual measurement $z(t+1)$ is trusted less, while the a priori estimate $\hat{x}(t+1|t)$ is trusted more.

However, when the noise is weaker, the measurement noise covariance matrix $\mathbf{R}(t+1)$ approaches 0. As a result, when the a posteriori estimate $\hat{x}(t+1|t+1)$ is calculated, the actual measurement $z(t+1)$ is almost trusted fully, while the a priori estimate $\hat{x}(t+1|t)$ is hardly trusted, thus failing to reduce noise effectively. To solve this problem, $R_{\min}$ is introduced as the lower bound for $\mathbf{R}(t+1)$:

$$R_{\min} = \alpha \tag{3-20}$$

where α is a quite small value. Once $\mathbf{R}(t+1)$ is less than $R_{\min}$, $\mathbf{R}(t+1)$ is set as $R_{\min}$ by default. Also, it is supposed that $\mathbf{R}(t+1) = R_{\min}$ within the initial period W.

Similarly, when the missing data exists (in this chapter, it is supposed that the actual measurement for the missing data is 0), $R_{\max}$ is defined as the upper bound for $\mathbf{R}(t+1)$ to enhance the trust in $\hat{x}(t+1|t)$:

$$R_{\max} = \beta \tag{3-21}$$

where β is a quite large value. Specifically, in the presence of missing data at time t+1, $\mathbf{R}(t+1)$ is set as $R_{\max}$ by default to recover the missing data much better. This practice is based on the fact that $z(t+1)$ is quite far from the real value at time t+1, hardly worth trust.

In general, the dynamic quadratic prediction model based Kalman filtering technique is

performed with (3-10)-(3-21). In the following section, the dynamic Kalman filtering technique will be combined with RMT for event detection.

3.5 Methodology Procedure

Due to the high sampling frequency of PMUs, PMU data, including voltage, current, and frequency, provides an enhanced snapshot of the dynamics of power systems, and thus much greater insight into power systems can be gathered from it after data conditioning, compared to SCADA data. Hence, in this chapter, as voltage does not need the network topology, voltage magnitude, pointing to more local phenomena, and voltage phase angle, pointing to more wide area behaviours, are selected to conduct real-time event detection respectively.

To perform real-time analysis, an $N \times T$ real-time split window is utilised to obtain the raw matrix $\mathbf{X}$ from the raw data source $\mathbf{\Omega}$. N denotes the number of state variables, namely, voltage magnitude and phase angle, and T denotes the time period. More specifically, at time t, the raw matrix $\mathbf{X}(t)$ is formed as follows:

$$\mathbf{X}(t) = [\mathbf{x}(t-T+1) \quad \mathbf{x}(t-T+2) \quad \cdots \quad \mathbf{x}(t-1) \quad \mathbf{x}(t)] \tag{3-22}$$

where $\mathbf{x}(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_N(t)]^{\mathrm{T}}$ is the measurement data at time t.

In order to add white Gaussian noise into the original data source $\mathbf{O}$ to obtain the raw data source $\mathbf{\Omega}$, the signal-to-noise ratio (SNR) is defined as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\mathrm{Tr}(\mathbf{O}\mathbf{O}^{\mathrm{T}})}{\mathrm{Tr}(\mathbf{G}\mathbf{G}^{\mathrm{T}})\times m^2}\right) \tag{3-23}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of matrix, $\mathbf{G}$ is the white Gaussian noise matrix, whose entries obey the standard normal distribution, m is the magnitude of white Gaussian noise, and $\mathbf{G}^{\mathrm{T}}$ and $\mathbf{O}^{\mathrm{T}}$ are the transposes of $\mathbf{G}$ and $\mathbf{O}$ respectively. Thus, during a study period $T_{\max}$, the raw data source $\mathbf{\Omega}$ is obtained through the following formulas:

$$\mathbf{\Omega} = \mathbf{O} + m\mathbf{G} \tag{3-24}$$

$$m = \sqrt{\frac{\mathrm{Tr}(\mathbf{O}\mathbf{O}^{\mathrm{T}})}{\mathrm{Tr}(\mathbf{G}\mathbf{G}^{\mathrm{T}})\times 10^{\mathrm{SNR}/10}}} \tag{3-25}$$

where $\mathbf{\Omega}$, $\mathbf{O}$, and $\mathbf{G}$ are all $N \times T_{\max}$ matrices.

In the proposed method, as the event indicator, MSR is calculated at each time and finally visualized to detect events. For simplicity, we set L=1 when forming the matrix product **Z**.

To mitigate against false positives, the dynamic Kalman filter also serves as the data conditioner. Thus, after data conditioning, the conditioned raw matrix $\widehat{\mathbf{X}}(t)$ is formed as follows:

$$\widehat{\mathbf{X}}(t) = [\hat{\mathbf{x}}(t-T+1) \quad \hat{\mathbf{x}}(t-T+2) \quad \cdots \quad \hat{\mathbf{x}}(t-1) \quad \hat{\mathbf{x}}(t)] \tag{3-26}$$

where $\hat{\mathbf{x}}(t) = [\hat{x}_1(t|t) \quad \hat{x}_2(t|t) \quad \cdots \quad \hat{x}_N(t|t)]^{\mathrm{T}}$ is the conditioned data at time t. As a result, the procedure of real-time event detection based on random matrix theory and Kalman filtering is shown in Table 3-1.

Table 3-1 Procedure of real-time event detection based on random matrix theory and Kalman filtering

Kalman Filter	1) Initialization 2) Calculate $\hat{x}_i(t\|t)$ using (3-10), $1 \le i \le N$ 3) Update $\mathbf{R}_i(t)$ using (3-19)-(3-21) 4) Update $\mathbf{K}_i(t)$ using (3-16)
Event Detector	5) Form $\widehat{\mathbf{X}}(t)$ using (3-26) 6) Transform $\widehat{\mathbf{X}}(t)$ to $\mathbf{X}_{\mathrm{s}}(t)$ using (3-6) 7) Transform $\mathbf{X}_{\mathrm{s}}(t)$ to $\mathbf{X}_{\mathrm{u}}(t)$ using (3-7) 8) Form $\mathbf{Z}(t)$ using (3-2) 9) Transform $\mathbf{Z}(t)$ to $\mathbf{Z}_{\mathrm{s}}(t)$ using (3-8) 10) Calculate MSR using (3-5) 11) If $t < T_{\max}$, repeat 2)-10) at the next time t+1; otherwise, go to 12) 12) Visualize MSR

3.6 Case Studies and Results

In this section, the improved event detection method is tested and compared with the original method using both simulation data and real PMU data. More specifically, a performance comparison is first made between the dynamic and original Kalman filters. After that, the improved event detection method is compared with the original one from the perspectives of both heavy noise and data loss. Besides, voltage magnitude (in p.u.) and phase angle (in rad) are used respectively to validate the feasibility and effectiveness of the improved event detection method.

In addition, the simulation data is generated based on the standard IEEE 118-bus system [113], while the real PMU data is from the GB transmission system. For event detection, there are generally two types of signals in power systems, namely, white Gaussian noise and small random fluctuations, and events and faults. It is also assumed that the SNRs of voltage magnitude and phase angle are the same in the specific situation.

For all cases, let $\mathbf{\Gamma} = 0.3\mathbf{I}$ ($\mathbf{I}$ is a 3×3 matrix of ones), $\mathbf{Q} = 1 \times 10^{-2}$, $\alpha = 1 \times 10^{-2}$, $\beta = 1 \times 10^{6}$, $W = 50$, $N = 118$, and $T = 240$.

3.6.1 Data Conditioning

1) Noise reduction

A comparison is performed between the dynamic and original Kalman filters. During the whole period $T_{\max} = 1500$, there is a discrete event (an increase of active power demand at bus 60) happening at $t = 601$. In this case, the constant $\mathbf{R}$ of the original Kalman filter is 1×10^{-2}. The results are shown in Table 3-2.

Table 3-2 Performance comparison of noise reduction between the dynamic and original Kalman filters ($T_{\max}$ = 1500)

SNR	RMSE of Voltage Magnitude			RMSE of Voltage Phase Angle		
	Raw Values	Original KF	Dynamic KF	Raw Values	Original KF	Dynamic KF
5	17.519	8.6245	3.7744	6.6207	4.7591	3.9543

15	5.6129	2.7867	2.3817	2.1212	1.0869	0.9555
25	1.7739	0.8944	0.8906	0.6704	0.3405	0.3385
35	0.5534	0.2807	0.2765	0.2091	0.1089	0.1070
45	0.1738	0.0886	0.0877	0.0657	0.0426	0.0424
55	0.0550	0.0278	0.0276	0.0208	0.0278	0.0277
65	0.0174	0.0090	0.0088	0.0066	0.0260	0.0259
75	0.0055	0.0029	0.0027	0.0021	0.0258	0.0257
85	0.0018	0.0010	0.0009	0.0007	0.0257	0.0257
95	0.0006	0.0006	0.0005	0.0002	0.0257	0.0257

In Table 3-2, root mean square error (RMSE) is used to make a performance comparison between the dynamic and original Kalman filters in terms of noise reduction. In order to eliminate the effect of the error at the beginning stage (namely initialization) on RMSE, RMSE is calculated during $t = 501 - 1500$. It is clearly shown that RMSE of voltage magnitude of the dynamic Kalman filter is almost the same as that of the original Kalman filter when SNR is greater than or equal to 25, while RMSE of voltage magnitude of the dynamic Kalman filter is less than that of the original Kalman filter when SNR is less than or equal to 15. Similarly, RMSE of voltage phase angle of the dynamic Kalman filter is almost the same as that of the original Kalman filter when SNR is greater than or equal to 25, while RMSE of voltage phase angle of the dynamic Kalman filter is less than that of the original Kalman filter when SNR is less than or equal to 15. In other words, when the noise in PMU data is heavy, the dynamic Kalman filter performs noise reduction much better than the original one, regardless of voltage magnitude or phase angle. However, when the noise is weak, there is no significant difference between these two. The reason for this practice is that the dynamic Kalman filter enhances the trust in the a posteriori estimate through adjusting the measurement noise covariance matrix **R** dynamically, therefore reducing noise more effectively, when there is heavy noise in PMU data.

2) Missing data recovery

A further comparison is performed between the dynamic and original Kalman filters in terms of missing data recovery, as shown in Table 3-3. In this case, the same discrete event happens to bus 60 as above, and RMSE is still calculated during $t = 501 - 1500$

and the constant **R** of the original Kalman filter remains 1×10^{-2}.

Table 3-3 Performance comparison of missing data recovery between the dynamic and original Kalman filters (T_{max} = 1500)

Percentage of Missing Data	RMSE of Voltage Magnitude (SNR = 95)			RMSE of Voltage Phase Angle (SNR = 35)		
	Raw Values	Original KF	Dynamic KF ($\times 10^{-4}$)	Raw Values	Original KF ($\times 10^{-2}$)	Dynamic KF ($\times 10^{-2}$)
5%	6.96	3.08	6	2.63	18.22	16.82
10%	10.02	4.62	7	3.71	18.27	16.84
15%	12.08	6.17	8	4.62	18.46	17.40
20%	14.12	7.50	9	5.33	18.63	17.87
25%	15.63	8.84	9	5.95	18.78	18.29
30%	17.17	10.37	10	6.52	18.90	18.79
35%	18.46	11.74	11	7.05	19.52	19.11
40%	19.85	12.96	12	7.53	19.88	19.68
45%	21.18	14.41	13	8.01	20.30	20.01
50%	22.11	16.03	13	8.42	20.82	20.71

In Table 3-3, as for voltage magnitude, it can be easily seen that RMSE of the dynamic Kalman filter is significantly lower than that of the original Kalman filter, no matter how much missing data there is. Besides, it is remarkable that as the percentage of missing data goes up, RMSE of the dynamic Kalman filter only increases slightly from 6×10^{-4} to 13×10^{-4}, while RMSE of the original Kalman filter increases dramatically from 3.08 to 16.03. The main reason for this fact is that the dynamic Kalman filter largely enhances the trust in the a posteriori estimate by setting the measurement noise covariance matrix **R** as a quite large value (e.g., in this case, it is 1×10^{6}), therefore weakening the trust in the measurement (namely 0) and recovering missing data more accurately.

However, it is shown clearly in Table 3-3 that RMSE of the dynamic Kalman filter is only slightly lower than that of the original Kalman filter in terms of voltage phase

angle. Moreover, when the percentage of missing data rises, both of them go up rather slowly. Specifically, the reason why the performance of the original Kalman filter is close to that of the dynamic one in terms of recovering voltage phase angle is as follows: when there is missing data at a certain time, the raw values of voltage magnitude and phase angle are both 0; however, the real value of voltage magnitude is much greater than that of voltage phase angle, and this leads to the fact that the difference of the real and raw values of voltage magnitude is much more significant than that of voltage phase angle; thus, for the particular constant **R** (e.g., in this case, $\mathbf{R} = 1 \times 10^{-2}$), the recovered value of voltage phase angle of the original Kalman filter is almost as close to the real value as that of the dynamic one. Furthermore, this is the very reason why the dynamic Kalman filter performs even better in terms of voltage magnitude recovery, compared with the original Kalman filter.

All in all, the dynamic Kalman filter outperforms the original one significantly, regardless of noise reduction or missing data recovery. This is the exact reason why the dynamic Kalman filter serves as the data conditioner of the proposed event detection method.

3.6.2 Event Detection Using Voltage Magnitudes

Event detection is conducted both in the situation with heavy noise and in the situation with missing data using voltage magnitudes. Different signals are used to test the performance of the improved and original event detection methods. The assumed signals are shown in Table 3-4.

Table 3-4 Assumed signals for event detection

Signal	Bus Number	Sampling Time	Active Power Demand (MW)
swell and sag signals	60	$t = 1 - 600$	fluctuation around 80
		$t = 601 - 700$	fluctuation around 120
		$t = 701 - 1050$	fluctuation around 80
		$t = 1051 - 1150$	fluctuation around 40
		$t = 1151 - 1500$	fluctuation around 80
	others	$t = 1 - 1500$	no change

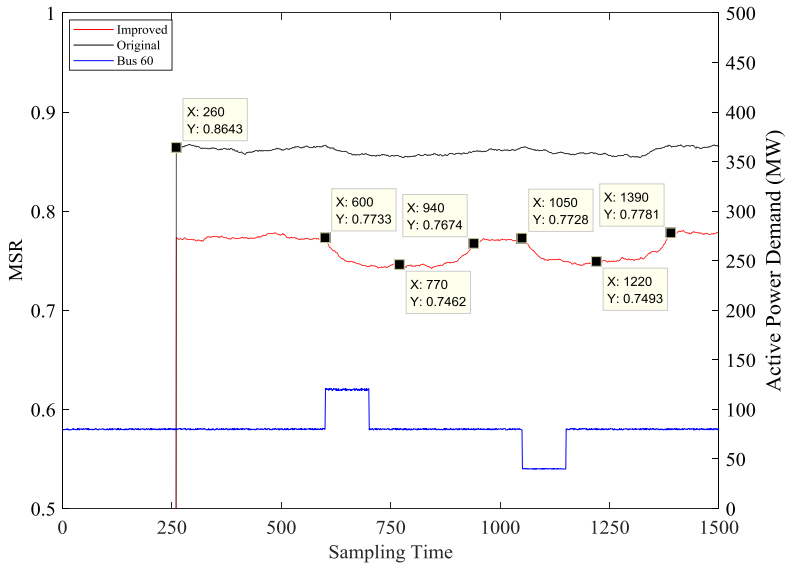
multiple signals	60	$t = 1 - 600$	fluctuation around 120
		$t = 601 - 700$	fluctuation around 150
		$t = 701 - 800$	fluctuation around 180
		$t = 801 - 900$	fluctuation around 210
		$t = 901 - 1000$	fluctuation around 240
		$t = 1001 - 1100$	fluctuation around 270
		$t = 1101 - 1200$	fluctuation around 300
		$t = 1201 - 1300$	fluctuation around 330
		$t = 1301 - 2000$	fluctuation around 360
	99	$t = 1 - 600$	fluctuation around 60
		$t = 601 - 700$	fluctuation around 100
		$t = 701 - 1200$	fluctuation around 60
		$t = 1201 - 1300$	fluctuation around 20
		$t = 1301 - 2000$	fluctuation around 60
	others	$t = 1 - 2000$	no change

1) Heavy noise

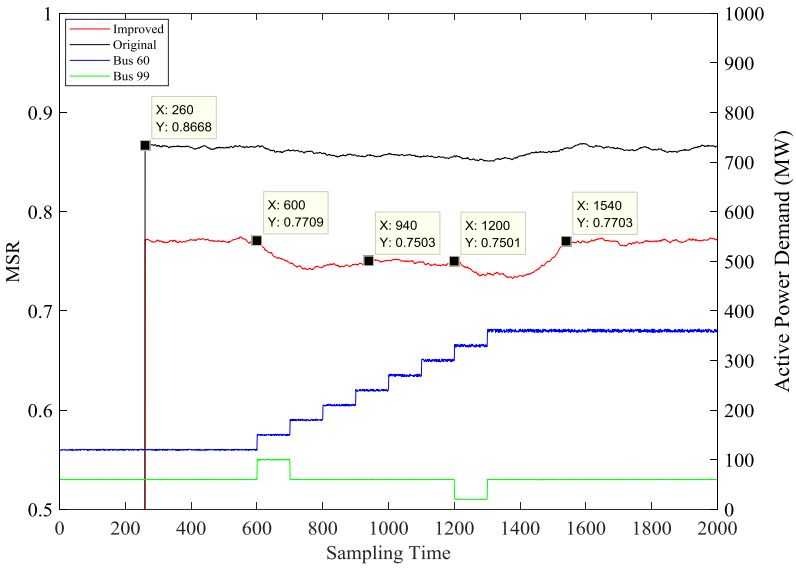
Figure 3-1 shows the MSR-*t* curves of the improved and original event detectors. It is noted that event detection starts after a short period, because the split window needs to be filled up with samples and the dynamic Kalman filter takes a short time to track the real values.

In Figure 3-1(a), according to the MSR-*t* curve of the improved event detector, signals can be detected easily. Specifically, during $t = 260 - 600$, MSR remains stable around 0.77, which means that the system state is normal without any signals. During this period, as there is only noise and an active power demand fluctuation, which play a dominant role in MSR, there is a slight fluctuation of MSR. Then, at $t = 600$, MSR starts to decrease gradually until $t = 770$ (from 0.7733 to 0.7462). After that, MSR begins to increase to the original stable level (0.7674) until $t = 940$, then remaining almost constant. During this period, a U-shaped curve is observed, which lasts for 340 instants. This indicates that there are signals changing the system state during $t = 601 - 940$. Besides, in this method, the effect of a signal on MSR extends *T* extra instants, due to historical data in the split window. So, the actual duration of the signals can be

calculated as $940 - 601 + 1 - T = 100$ instants, and the occurrence time of the signals is the starting time of the U-shaped curve, which is $t = 601$. That is exactly correspondent to the fact that there is an increase of active power demand at bus 60 during $t = 601 - 700$. As a result, event detection is conducted in this way. Likewise, there is a second U-shaped curve from $t = 1051$ to $t = 1390$, which indicates that the signals occur at $t = 1051$, and the actual duration of the signals is $1390 - 1051 + 1 - T = 100$ instants.



(a) Swell and sag signals (SNR = 75)



(b) Multiple signals (SNR = 70)

Figure 3-1 MSR-*t* curves of the improved and original event detectors in the situation with heavy noise

However, as for the original event detector, signals cannot be detected under the same situation. In other words, there is no clear U-shaped curve during the whole period. MSR changes slightly around 0.86 during $t = 600 - 1390$. This cannot be identified from the fluctuation of MSR, due to the fluctuation of active power demand and heavy noise. Hence, the original event detector is unable to discover any signals, when there is heavy noise in PMU data.

Apart from single signals, multiple signals can be also discovered by the improved event detector in the situation with heavy noise, as shown in Figure 3-1(b). According to the MSR-*t* curve of the improved event detector, two types of signals can be detected as follows: firstly, two U-shaped curves are found during $t = 601 - 940$ and $t = 1201 - 1540$, and it indicates continuous signals during $t = 601 - 700$ and $t = 1201 - 1300$ respectively; secondly, a third U-shaped curve is found during $t = 601 - 1540$, which indicates continuous signals during $t = 601 - 1300$. However, the same conclusion cannot be drawn from the MSR-*t* curve of the original event detector.

The reason why the improved event detector is still able to discover signals even in the situation with heavy noise can be explained as follows: as the data conditioner of the improved event detector, the dynamic Kalman filter can reduce noise in PMU data effectively, enabling it to discriminate signals from noise and fluctuations; thus, it can perform event detection better. However, as the original event detector does not have a data conditioner to reduce noise, it is unable to identify signals from noise and fluctuations, when there is heavy noise in PMU data.

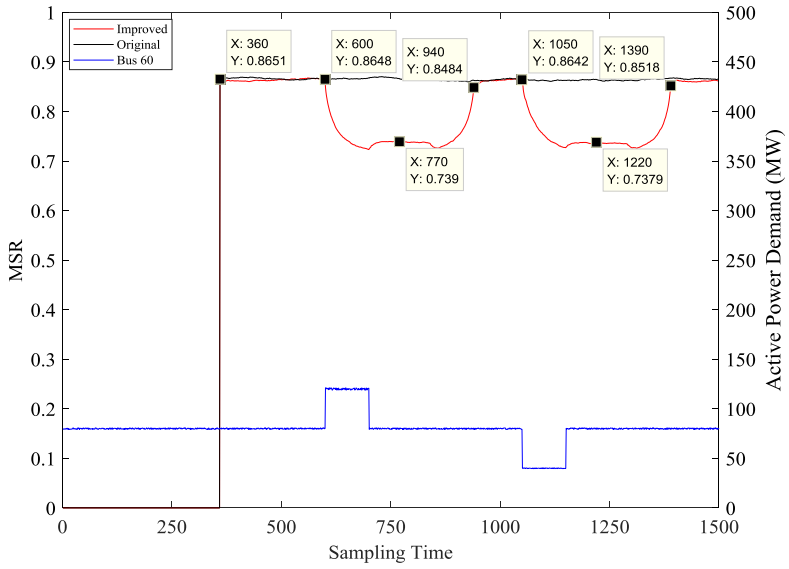
2) Missing data

Figure 3-2 depicts the MSR-*t* curves of the improved and original event detectors when the percentage of missing data is 30%. As the dynamic Kalman filter also needs to take a short time to track the real values in the situation with missing data, event detection starts after a short period.

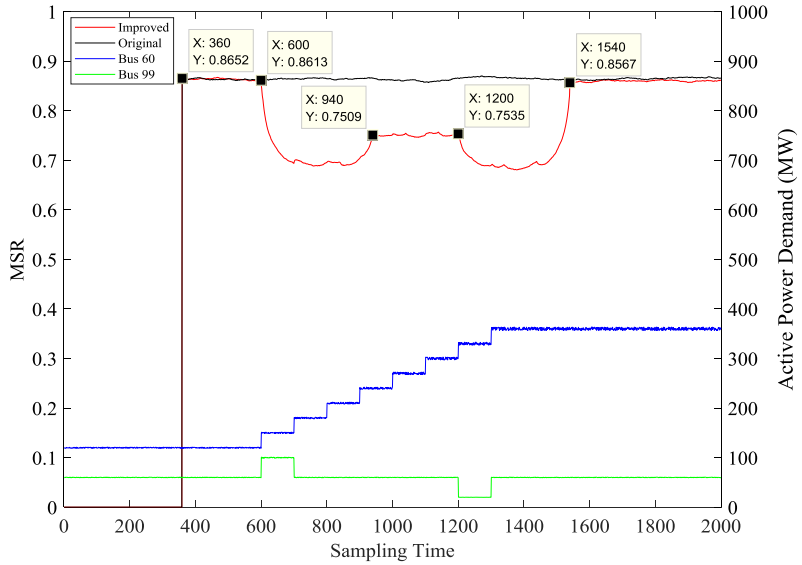
In Figure 3-2(a), according to the MSR-*t* curve of the improved event detector, signals can be detected in the same way mentioned above. To be specific, during $t = 360 -$

600, MSR remains stable around 0.86, which means that there is no signal in the system. During this period, as noise and an active power demand fluctuation play a dominant role, MSR fluctuates slightly around the stable level. Then, at $t = 600$, MSR starts to decline fast until $t = 770$ (from 0.8648 to 0.739). Afterwards, MSR begins to rise to the original stable level (0.8484) until $t = 940$, then remaining almost constant. During this period, a U-shaped curve is observed between $t = 601$ and $t = 940$, which indicates that there are signals changing the system state during $t = 601 - 940$. However, due to the extension effect of a signal on MSR, the actual duration of the signals is $940 - 601 + 1 - T = 100$ instants, and the occurrence time of the signals is $t = 601$, which is exactly the starting time of the U-shaped curve. Furthermore, it can be seen easily that this inference is correspondent to the fact that there is an increase of active power demand of bus 60 during $t = 601 - 700$. Similarly, a second U-shaped curve can be observed from $t = 1051$ to $t = 1390$, which suggests that there are signals occurring at $t = 1051$ and lasting for $1390 - 1051 + 1 - T = 100$ instants.

However, the original event detector is unable to detect signals under the same situation. That is to say, U-shaped curves cannot be found during the signal occurrence period. It can be easily seen from the MSR-*t* curve of the original event detector that MSR remains almost constant with a tiny fluctuation around 0.86 throughout the whole period. So, the original event detector is incapable to discover signals, when there are missing samples in PMU data.



(a) Swell and sag signals (SNR = 95)



(b) Multiple signals (SNR = 90)

Figure 3-2 MSR-*t* curves of the improved and original event detectors in the situation with missing data

In addition, as shown in Figure 3-2(b), multiple signals can be still discovered by the improved event detector in the situation with missing data. According to the MSR-*t* curve of the improved event detector, two types of signals can be detected as follows: firstly, two U-shaped curves are found during $t = 601 - 940$ and $t = 1201 - 1540$, which indicates continuous signals during $t = 601 - 700$ and $t = 1201 - 1300$ respectively; secondly, a third U-shaped curve is found during $t = 601 - 1540$, indicating continuous signals during $t = 601 - 1300$. Here, it is remarkable that MSR remains around 0.75 with a slight fluctuation between $t = 940$ and $t = 1200$. During this period, MSR is lower than the stable level (appropriately 0.86), but higher than the bottom level (appropriately 0.69), because there are only signals of bus 60 during $t = 940 - 1200$, without signals of bus 99. By contrast, the same signals cannot be discovered by the original event detector.

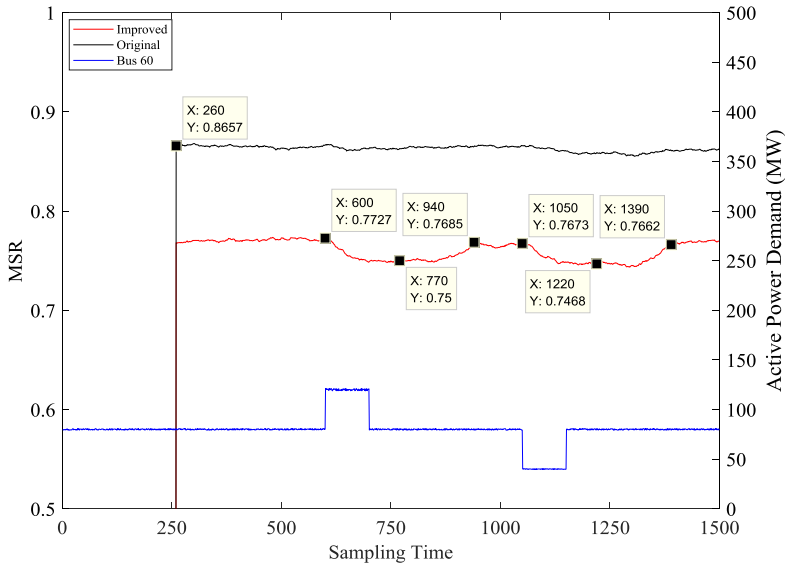
Based on the analysis above, it is obvious that the improved event detector is still capable to discover signals in the situation with missing data, while the original event detector fails to discover any signals. The reason for this is that the dynamic Kalman filter can recover missing samples in PMU data accurately, therefore enabling the improved event detector to identify signals. However, without a data conditioner, the original event detector is unable to recover missing samples and discover signals.

3.6.3 Event Detection Using Voltage Phase Angles

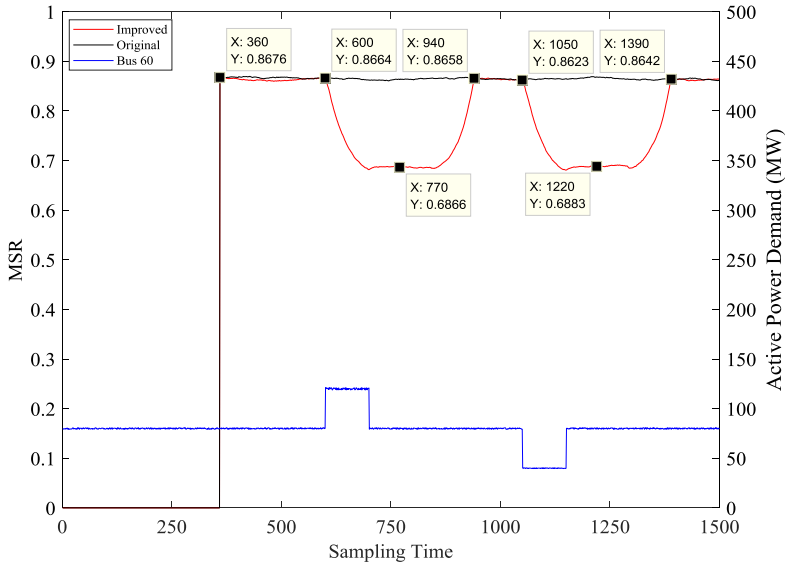
Event detection is also conducted using voltage phase angles (with respect to the slack bus) to further compare the improved and original event detection methods, both in the situation with heavy noise and in the situation with missing data. However, as voltage magnitude based event detection is quite similar to voltage phase angle based event detection, only swell and sag signals (as shown in Table 3-4) are employed to test both methods.

Figure 3-3(a) describes the MSR-*t* curves of the improved and original event detectors when SNR is 20, while Figure 3-3(b) shows the MSR-*t* curves of the improved and original event detectors when SNR is 35 and the percentage of missing data is 30%. In Figure 3-3(a) and Figure 3-3(b), regardless of heavy noise or missing data, a U-shaped curve can be observed from $t = 601$ to $t = 940$ and from $t = 1051$ to $t = 1390$ respectively for the improved event detector. By contrast, as for the original event detector, there is no U-shaped curve found during the whole period in both situations.

Based on the above results, the improved event detector conducts event detection still much better than the original one, even using voltage phase angles. The reason for this fact is that the dynamic Kalman filter is also capable to condition voltage phase angle effectively, apart from voltage magnitude.



(a) Swell and sag signals (in the situation with heavy noise)



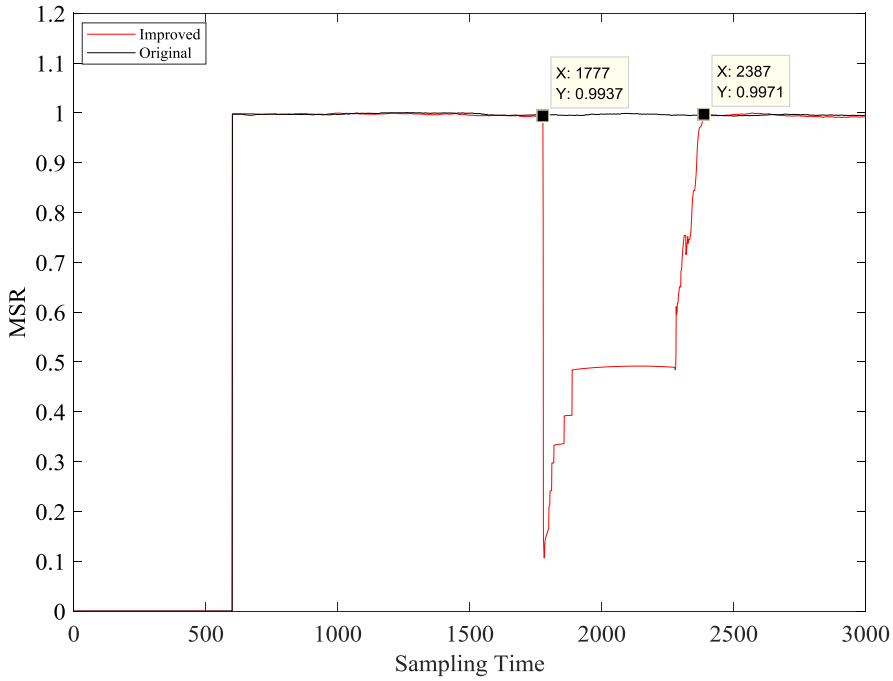
(b) Swell and sag signals (in the situation with missing data)

Figure 3-3 MSR-*t* curves of the improved and original event detectors using voltage phase angles

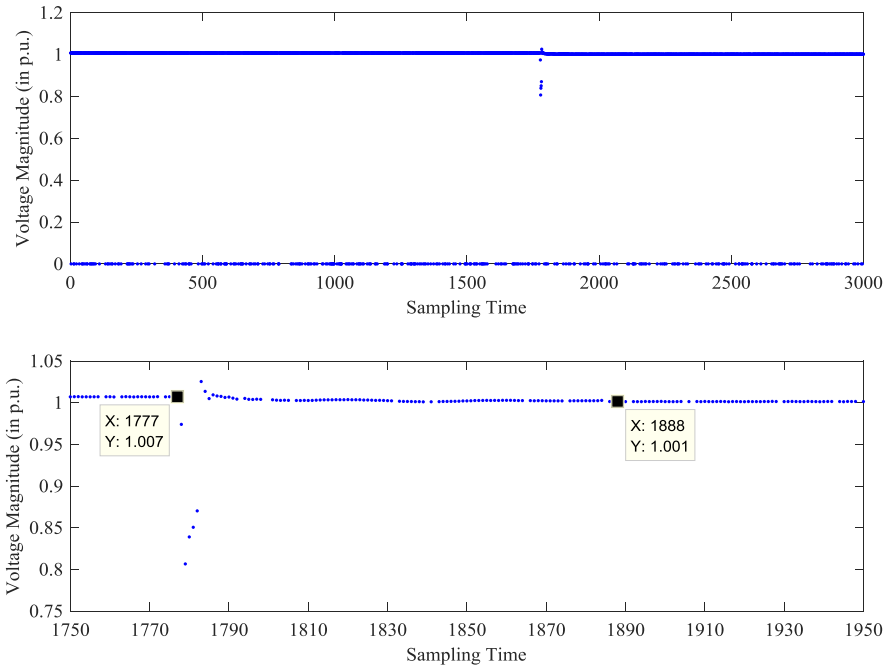
3.6.4 Event Detection Using Real PMU Data

Real PMU data is utilised to further verify the effectiveness and robustness of the proposed event detection method. The PMU data is from 13 PMUs of the GB transmission system [114-115], and the sampling rate of the PMUs is 50Hz. Besides, the total time length of the PMU data is 1 minute, and thus there are totally $50 \times 60 = 3000$ samples. What is more, the missing samples in the PMU data take up 12.5%. In this case, N is 13 and T is 500. The results are shown in Figure 3-4 and Figure 3-5.

In Figure 3-4(a) and Figure 3-5(a), it can be clearly observed that there is a U-shaped curve during $t = 1777 - 2387$ in the MSR-t curves of the improved event detector. This indicates that there are signals occurring at $t = 1778$ and lasting for $2387 - 1778 + 1 - T = 110$, which is $110/50 = 2.2$ seconds. According to Figure 3-4(b) and Figure 3-5(b), the voltage magnitude and phase angle of a certain PMU change dramatically during $t = 1778 - 1887$, which confirms the above inference. In fact, there is indeed a tripping event from $t = 1778$ to $t = 1887$ across the GB transmission system. However, the original event detector fails to detect this event, because no U-shaped curve can be seen throughout the whole period.

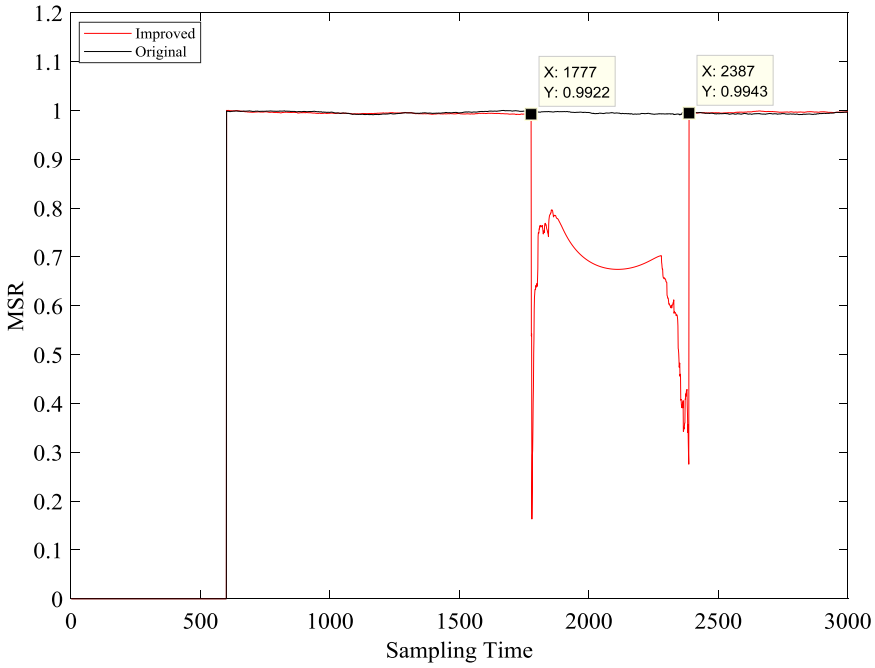


(a) MSR-*t* curves of the improved and original event detectors

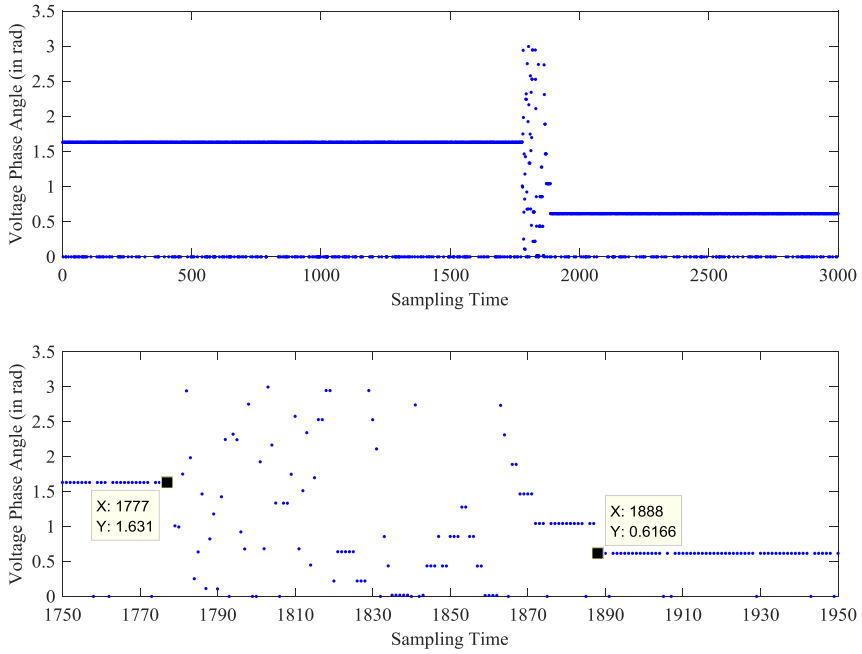


(b) Voltage magnitude of a certain PMU

Figure 3-4 Event detection using voltage magnitudes of real PMU data



(a) MSR-*t* curves of the improved and original event detectors



(b) Voltage phase angle of a certain PMU

Figure 3-5 Event detection using voltage phase angles of real PMU data

3.7 Chapter Summary

This chapter presented a real-time data driven event detection method based on random

matrix theory and Kalman filtering. In addition, as the data conditioner of the presented method, a dynamic Kalman filtering technique was developed through the adjustment of the measurement noise covariance matrix in order to reduce noise and recover missing samples in PMU data. The experimental results have shown that the dynamic Kalman filter outperforms the original one, in terms of both noise reduction and missing data recovery. Furthermore, the comparison results have shown that the improved event detection method is much more robust than the original one, especially in practical situations where PMU data is collected with significantly heavy noise and plenty of missing samples.

Chapter 4

Short-Term Residential Load Forecasting Based on Deep Learning and K-means Clustering

4.1 Introduction

As the modern power network is transitioning gradually towards the smart grid with high penetration of renewable energy generation, load forecasting is playing an increasingly important role in planning and operation of the future power network [116-117]. A good understanding of electricity consumption patterns can also bring a great many economic benefits to power generators and network operators as well as electricity customers. Apparently, the rapid deployment of advanced metering infrastructure (AMI) across the power network, for example, smart meters, has already generated a huge amount of monitoring data and significantly facilitated load forecasting over the past few years.

In general, load forecasting is divided into network-level (or system-level) load forecasting and residential load forecasting. During recent years, a number of researchers have mainly focused on network-level load forecasting methods. To be specific, these methods include support vector regression (SVR) [118-121], fuzzy inductive reasoning (FIR) [122], artificial neural networks (ANNs) [123-130], and deep neural networks (DNNs) [131-135]. All these methods have achieved a high prediction accuracy on network-level load forecasting, in particular, DNNs. However, as residential load is much more volatile and uncertain than network-level load, traditional forecasting methods, which are quite suitable for network-level load, cannot obtain a satisfactory prediction accuracy on residential load. As a result, a few researchers have attempted to apply deep learning to residential load forecasting [55, 136], for example, LSTM networks. But, deep learning methods need a huge amount of data to build a well-trained model, otherwise tending to incur overfitting.

Most recently, a novel residential load forecasting method was presented based on deep learning [137]. It firstly divides residential customers into groups to generate a load profile pool for each group and then applies an LSTM network to build a forecasting model for each group, finally performing load forecasting for each resident. Obviously, this method takes advantage of load profile pools to overcome overfitting during training and enables forecasting models to learn correlation and interaction among residential customers. However, it is unable to sufficiently extract similarity within residential load of a customer group, because it simply divides residential customers

into groups in a random way, easily resulting in significant differences in a load profile pool.

Thus, this chapter develops a short-term individual residential load forecasting method on the basis of deep learning and k-means clustering, which is capable to extract similarity of residential load more effectively and perform load forecasting more accurately at the individual residential level. Specifically, it makes full use of k-means clustering to extract similarity among residential load and deep learning to extract complicated patterns of residential load. In addition, in order to improve the forecasting accuracy, a comprehensive feature expression strategy is utilised to describe load characteristics of each time step in detail. Finally, the developed short-term residential load forecasting method is demonstrated on a real-life residential load dataset from Ireland. The experimental results suggest that it can achieve a higher forecasting accuracy in terms of both RMSE and MAE.

4.2 Overview of Load Forecasting

There have been a large number of research works in the field of load forecasting at the network level. In reference [118], a new version of SVR was presented to predict network load, based on the modification of the risk function of SVR with using locally weighted regression while keeping the regularization term in its original form. Besides, the weighted distance algorithm, based on the Mahalanobis distance for optimizing the bandwidth of the weighting function, was proposed to improve the accuracy. Reference [121] developed a short-term multiregion load forecasting method for day ahead operation on the basis of SVR and weather and load diversity. The developed method is able to find the optimal region partition according to the load and weather characteristics and then generate more accurate forecasts for aggregated system load. Different from reference [118] and reference [121], reference [122] applied FIR to short-term load forecasting for a day in advance. The FIR model learns both past and future relations from load and temperature, and the proposed optimization model utilises an evolutionary algorithm based on a local random controlled search, simulated rebounding algorithm, to select the inputs for the FIR model.

Besides, ANNs have been applied widely in the research area of network-level load

forecasting. For example, Chen presented a similar day based wavelet neural network method to predict system load for the next day [123]. It selects the load of the similar day as input and applies wavelet to decompose it into a low-frequency component and a high-frequency component, finally using separate neural networks to predict the two components of the load for the next day. Similarly, Nima proposed a hybrid method for short-term bus load forecasting, which combines a forecast-aided state estimator (FASE) and a multilayer perceptron (MLP) neural network [124]. The FASE forecasts the hourly load of each bus by the means of its previous data, and then the inputs and outputs of the FASE are fed to the MLP neural network to generate the final forecast for the bus load. Another short-term load forecasting method was developed for the distribution level in reference [127]. Within this method, the load of the root node of any user-defined subtree is firstly predicted through a wavelet neural network with appropriate inputs. Then, the load of its child nodes categorised as regular and irregular based on load pattern similarity is predicted separately. In addition, James investigated the use of weather ensemble predictions in the application of ANNs to load forecasting due to the importance of weather forecasts for load forecasting [129]. A weather ensemble prediction is comprised of multiple scenarios for a weather variable, which are employed to produce multiple load scenarios in order to estimate the uncertainty in the ANN load forecast and improve the prediction accuracy.

Apart from applications of traditional machine learning to network-level load forecasting, many researchers have also attempted to apply deep learning to predict network load. Jatin proposed an empirical mode decomposition (EMD) based deep learning method, combining the EMD method with LSTM networks to estimate the electricity demand for the given season, day, and time interval of a day [133]. The EMD method decomposes a load time series signal into several intrinsic mode functions (IMFs) and residual, and then the LSTM networks are trained separately for each of the extracted IMFs and residual. Finally, the prediction results of all IMFs are summed to determine the aggregated output for the electricity demand. In reference [134], a short-term load forecasting model was presented based on a modified deep residual network, and a two-stage ensemble strategy was utilised to enhance its generalization capacity in order to improve the prediction results. Similarly, reference [135] employed a time-dependency CNN and a cycle based LSTM network respectively for short-term and medium-term load forecasting.

As for load forecasting at the residential level, only a few efforts have been made by researchers over recent years, mainly focusing on deep learning. Kong explored the application of LSTM networks to short-term residential load forecasting for a single electricity customer [136]. A comprehensive comparison is made to demonstrate that the LSTM model outperforms various benchmark models. Likewise, a residential load forecasting method was proposed in reference [55], which combines LSTM networks with resident behaviour learning. The proposed method uses both aggregated consumption data and appliance consumption sequences as inputs to train the LSTM networks. Although DNNs have a superior capability to extract complex patterns of residential load, they are prone to overfitting without enough training data. Hence, in order to avoid overfitting, reference [137] designed a novel pooling deep RNN (PDRNN) to perform residential load prediction. However, in spite of its ability to overcome overfitting, this method only separates residential load profiles into groups randomly, therefore likely to cause significant differences within the load profiles of each group and affect its prediction performance negatively. Considering these drawbacks, this chapter proposes a solution to them.

The remainder of this chapter is organized as follows. Section 4.3 formulates LSTM networks in detail. Section 4.4 introduces k-means clustering briefly. Section 4.5 details the combination of deep learning with k-means clustering for short-term residential load forecasting. Section 4.6 uses a real-life Irish residential load dataset to evaluate the developed short-term residential load forecasting method. Finally, the conclusions are drawn in Section 4.7.

4.3 Long Short Term Memory

As a sequence based model to deal with time series problems, RNNs are fundamentally different from conventional feedforward neural networks. They are capable to establish excellent temporal correlation between previous and current information, which means that the decision an RNN makes at time step t-1 could influence the decision it makes at time step t. This characteristic makes RNNs an ideal candidate for short-term residential load forecasting, because the residential load consumption pattern has a strong and complex relationship between adjacent time steps [136]. However, in terms of the

specific implementation, a special RNN, called the LSTM network, is employed in this chapter, as it significantly improves the performance of the general RNN. So, in this section, the RNN architecture is introduced in the first place, and then the LSTM unit is explained.

4.3.1 Recurrent Neural Networks

In RNNs, the shared states are decomposed into multiple RNN layers for the purpose of gaining the desirable properties of RNNs. In Figure 4-1, the computational graph and the unfolded topological graph of an *N* layer RNN are shown to demonstrate the working process of RNNs [137].

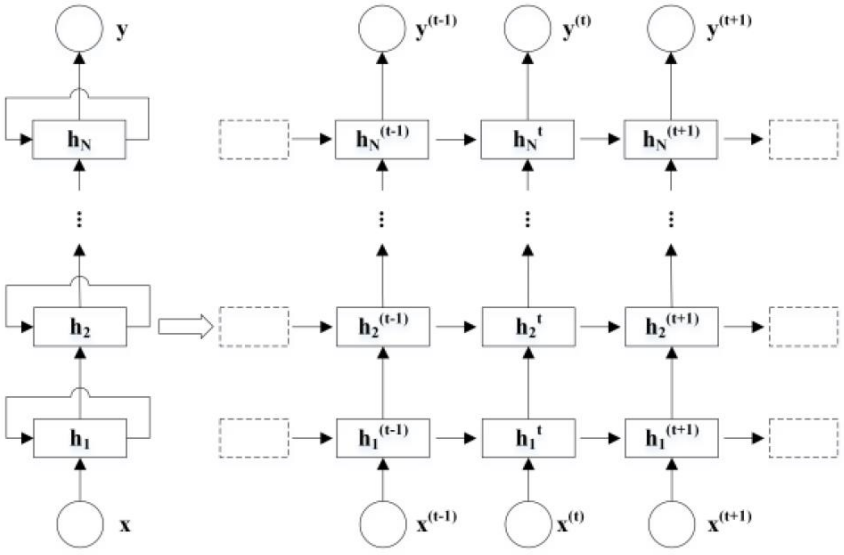


Figure 4-1 Computational graph and unfolded topological graph of an *N* layer RNN [137]

In the computational graph, the RNN aims to map the input sequence of x into corresponding sequential outputs y. As presented in the computational graph, the learning process conducts every single time step from $t = 1$ to $t = \tau$. For time step t, the network neuron parameters at the l^{th} layer update their shared states with the following equations [137]:

$$a_1^{(t)} = b_1 + W_1 \cdot h_1^{(t-1)} + U_1 \cdot x^{(t)} \tag{4-1}$$

$$h_l^{(t)} = f_{\text{activation}}\left(a_l^{(t)}\right) \qquad (l = 1, 2, \cdots, N) \tag{4-2}$$

$$a_l^{(t)} = b_l + W_l \cdot h_l^{(t-1)} + U_l \cdot h_{l-1}^{(t)} \qquad (l = 2, 3, \cdots, N) \tag{4-3}$$

$$y^{(t)} = b_N + W_N \cdot h_N^{(t-1)} + U_N \cdot h_N^{(t)} \tag{4-4}$$

$$L = f_{\text{loss}}\left(y^{(t)}, y_{\text{target}}^{(t)}\right) \tag{4-5}$$

where $x^{(t)}$ is the data input at time step t, $y^{(t)}$ is the corresponding forecast, $y_{\text{target}}^{(t)}$ is the true output at time step t, $h_l^{(t)}$ is the shared states of the l^{th} layer at time step t, and $a_l^{(t)}$ is the input of the l^{th} layer at time step t, which consists of three components: 1) the input $x^{(t)}$ at time step t or the shared states $h_{l-1}^{(t)}$ of the l-1$^{\text{th}}$ layer at time step t; 2) the bias b_l of the l^{th} layer; 3) the shared states $h_l^{(t-1)}$ of the l^{th} layer at time step t-1. Due to their shared states, RNNs are able to learn uncertainty repeated in previous time steps.

4.3.2 LSTM Units

RNNs are trained by backpropagation through time, but learning long-term dependency with RNNs is very difficult because of gradient vanishing or exploding [136]. Specifically, gradient vanishing refers to the exponential decrease of the norm of the gradient for long-term components to zero, therefore limiting the ability of RNNs to learn long-term temporal correlation, while gradient exploding refers to the opposite case. In order to overcome these two issues, the LSTM unit was firstly introduced, which included a memory cell, and further improved with an extra forget gate. As a result, the LSTM network has become the most popular structure of RNNs in many time series problems.

Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_T\}$ denote a typical input sequence for an LSTM unit, where $\boldsymbol{x}_t \in \mathbf{R}^k$ represents a k-dimensional vector of real values at time step t. In order to establish temporal connections, the LSTM unit defines and maintains an internal memory cell state throughout the whole life cycle, which is the most important element of the LSTM unit. The memory cell state $\boldsymbol{s}_{t\text{-}1}$ interacts with the intermediate output $\boldsymbol{h}_{t\text{-}1}$ and the subsequent input $\boldsymbol{x}_t$ to determine which elements of the internal state vector should be updated, maintained, or erased according to the outputs of the previous time step and the inputs of the present time step. In addition to the internal state, the LSTM unit also defines the input node $\boldsymbol{g}_t$, the input gate $\boldsymbol{i}_t$, the forget gate $\boldsymbol{f}_t$, and the output gate $\boldsymbol{o}_t$. The

formulations of all nodes in the LSTM unit are given below from (4-6) to (4-11) [136]:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f\right) \tag{4-6}$$

$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i\right) \tag{4-7}$$

$$\boldsymbol{g}_t = \Phi\left(\boldsymbol{W}_{gx}\boldsymbol{x}_t + \boldsymbol{W}_{gh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_g\right) \tag{4-8}$$

$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_{ox}\boldsymbol{x}_t + \boldsymbol{W}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_o\right) \tag{4-9}$$

$$\boldsymbol{s}_t = \boldsymbol{g}_t \odot \boldsymbol{i}_t + \boldsymbol{s}_{t-1} \odot \boldsymbol{f}_t \tag{4-10}$$

$$\boldsymbol{h}_t = \Phi(\boldsymbol{s}_t) \odot \boldsymbol{o}_t \tag{4-11}$$

where $\boldsymbol{W}_{fx}$, $\boldsymbol{W}_{fh}$, $\boldsymbol{W}_{ix}$, $\boldsymbol{W}_{ih}$, $\boldsymbol{W}_{gx}$, $\boldsymbol{W}_{gh}$, $\boldsymbol{W}_{ox}$, and $\boldsymbol{W}_{oh}$ are the weight matrices of the corresponding inputs of the network activation functions, $\odot$ denotes the element-wise multiplication, σ denotes the sigmoid activation function, and Φ denotes the tanh activation function. The structure of an LSTM unit at a single time step is illustrated in Figure 4-2, and the structure of an unrolled LSTM unit at consecutive time steps is shown in Figure 4-3 [136]. It is noted that the weights of an unrolled LSTM unit are duplicated at every time step.

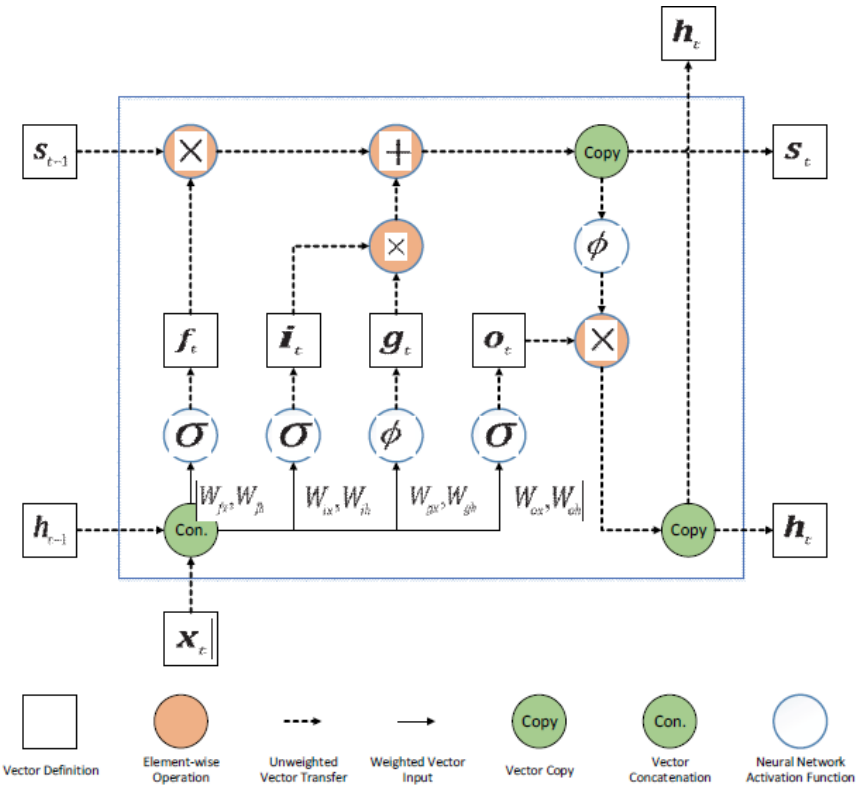


Figure 4-2 Structure of an LSTM unit [136]

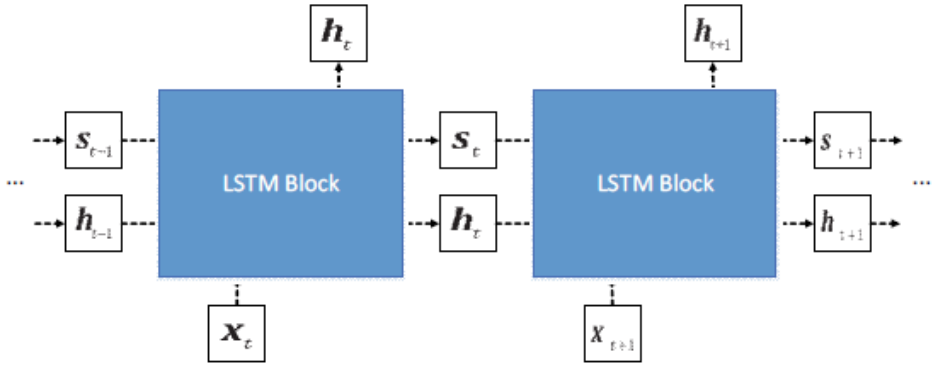


Figure 4-3 Structure of an unrolled LSTM unit [136]

In order to train a simple one layer LSTM network, the dimension n of the hidden output should be specified. In this case, the hidden output $\boldsymbol{h}_t \in \mathbf{R}^n$ at a given time step is an n-dimensional vector. Accordingly, $\boldsymbol{s}_t$ is also n-dimensional. In general, both $\boldsymbol{h}_t$ and $\boldsymbol{s}_t$ are initialised as zero, namely, $\boldsymbol{h}_t = \mathbf{0}$ and $\boldsymbol{s}_t = \mathbf{0}$. Besides, in an LSTM unit, there are three sigmoid functions with the output range from 0 to 1, serving as the soft switches to determine which signals should pass the gates. The decisions of the input gate $\boldsymbol{i}_t$, the forget gate $\boldsymbol{f}_t$, and the output gate $\boldsymbol{o}_t$ are all dependent on the current input $\boldsymbol{x}_t$ and the previous output $\boldsymbol{h}_{t-1}$. The input gate controls what to preserve in the internal state, while the forget gate controls what to forget from the previous state $\boldsymbol{s}_{t-1}$. With the internal state updated, the output gate decides which the internal state $\boldsymbol{s}_t$ should pass as the LSTM output $\boldsymbol{h}_t$. This process then repeats at the next time step. All the weights and biases are updated through minimizing the difference between the final LSTM outputs and the actual output values. Based on this unrolled structure, the information of the current time step can be stored and maintained to affect the LSTM outputs of the future time steps.

4.4 K-means Clustering

Clustering is a common unsupervised learning task, which is intended for identifying the clusters of the similar data samples in a dataset [138]. Although numerous clustering methods have been developed, k-means clustering is still one of the most popular clustering methods in many fields due to its simplicity and robustness. As a result, k-means clustering is adopted to cluster residential load profiles in this chapter.

Given a dataset $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbf{R}^{p \times n}$, the goal of k-means clustering is to partition the data into K clusters so that $\boldsymbol{\mu}_k \in \mathbf{R}^p$ is the prototype associated with the k^{th} cluster for $k = 1, 2, \cdots, K$. A set of binary indicator variables $c_{ik} \in \{0, 1\}$ are also introduced to represent the assignments, where $\mathbf{c}_i = [c_{i1}, c_{i2}, \cdots, c_{iK}]^{\mathrm{T}}$ is the k^{th} canonical basis vector in $\mathbf{R}^K$ if and only if $\mathbf{x}_i$ belongs to the k^{th} cluster.

Let $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^K$ be the cluster centers and $\mathbf{c} = \{\mathbf{c}_i\}_{i=1}^n$ be the assignments of the data samples. K-means clustering attempts to minimize the sum of the squared Euclidean distances of all the data points to their assigned clusters [138]:

$$\boldsymbol{J}(\mathbf{c}, \boldsymbol{\mu}) = \sum_{i=1}^n \sum_{k=1}^K c_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_2^2 \tag{4-12}$$

where the objective function $\boldsymbol{J}(\mathbf{c}, \boldsymbol{\mu})$ is minimized by respectively updating the assignments $\mathbf{c}$ and the cluster centers $\boldsymbol{\mu}$ in an iterative way. The update procedure is formulated as follows [138]:

Step 1: Minimize $\boldsymbol{J}(\mathbf{c}, \boldsymbol{\mu})$ over $\mathbf{c}$ while keeping $\boldsymbol{\mu}$ fixed:

$$\forall i = 1, 2, \cdots, n: \quad c_{ik} = \begin{cases} 1 & k = \operatorname{argmin}_j \left\|\mathbf{x}_i - \boldsymbol{\mu}_j\right\|_2^2 \\ 0 & \text{otherwise} \end{cases} \tag{4-13}$$

Step 2: Minimize $\boldsymbol{J}(\mathbf{c}, \boldsymbol{\mu})$ over $\boldsymbol{\mu}$ while keeping $\mathbf{c}$ fixed:

$$\forall k = 1, 2, \cdots, K: \quad \boldsymbol{\mu}_k = \frac{1}{n_k} \sum_{i \in \mathrm{S}_k} \mathbf{x}_i \tag{4-14}$$

where S_k denotes the set of the data samples assigned to the k^{th} cluster, and n_k denotes the number of the data samples S_k contains.

Generally, in order to improve the performance of k-means clustering, the k-means++ algorithm is applied to initialise the cluster centers. Hence, the whole process of k-means clustering is shown in Table 4-1 [139].

Table 4-1 Process of k-means clustering

1) Initialise the *k* cluster centers by k-means++
2) Assign each data sample to the closest cluster center using (4-13)
3) Update each cluster center using (4-14)

4) If no cluster members are reassigned to the new clusters or the decrease of the objective function $\boldsymbol{J}(\mathbf{c}, \boldsymbol{\mu})$ is within the predefined limit, stop iterating; otherwise, repeat 2)-3)

4.5 Implementation Procedure

In this section, the methodological implementation of the proposed method, the experiment dataset, and the experiment setup are introduced in detail.

4.5.1 Methodological Implementation

As residential load is rather volatile and uncertain, a comprehensive feature description strategy is applied in this chapter to improve the forecasting accuracy. According to this strategy, the input features of a data sample $\mathbf{S}_t$ at a particular time step t are explained as follows [136-137]:

1) the sequence of the residential load for the past T time steps $\mathbf{E}_t = [e_{t-T}, \cdots, e_{t-2}, e_{t-1}] \in \mathbf{R}^T$;

2) the sequence of the time indexes for the past T time steps $\mathbf{D}_t \in \mathbf{R}^T$, where the range for each element is [1, 48], because the sampling frequency is once every half an hour;

3) the sequence of the day indexes for the past T time steps $\mathbf{W}_t \in \mathbf{R}^T$, where each element ranges from 1 to 7, as there are 7 days in a week;

4) the sequence of the holiday signs for the past T time steps $\mathbf{H}_t \in \mathbf{R}^T$, where each element is either 1 or 2, and 1 denotes non-holiday and 2 denotes holiday (in this chapter, it is assumed that weekdays are non-holiday and weekends are holiday);

5) the sequence of the customer indexes in the corresponding cluster $\mathbf{P} \in \mathbf{R}^T$, where each element is the same and belongs to [1, L], and L is the number of the customers of the cluster.

So, a data sample $\mathbf{S}_t = \left[\mathbf{E}_t^{\mathrm{T}}, \mathbf{D}_t^{\mathrm{T}}, \mathbf{W}_t^{\mathrm{T}}, \mathbf{H}_t^{\mathrm{T}}, \mathbf{P}_t^{\mathrm{T}}\right]$ is a matrix of a concatenation of the five sequences. In order to speed up the convergence of the LSTM network and improve its generalization capacity, the input features are standardized to [0, 1] according to their nature. Specifically, the min-max standardization method is adopted for $\mathbf{E}_t$, while $\mathbf{D}_t$,

$\mathbf{W}_t$, $\mathbf{H}_t$, and $\mathbf{P}$ are encoded by a one-hot encoder. The one-hot encoder maps an original element of the feature sequence with M categories into a new sequence with M elements, where only the new element corresponding to the original element is one while the rest are all zero. Therefore, a standardized data sample $\widehat{\mathbf{S}}_t = \left[\widehat{\mathbf{E}}_t^{\mathrm{T}}, \widehat{\mathbf{D}}_t^{\mathrm{T}}, \widehat{\mathbf{W}}_t^{\mathrm{T}}, \widehat{\mathbf{H}}_t^{\mathrm{T}}, \widehat{\mathbf{P}}_t^{\mathrm{T}}\right]$ is a $T \times N$ matrix, where $N = 1 + 48 + 7 + 2 + L$. Each row of the standardized input sample $\widehat{\mathbf{S}}_t$ is the detailed features for the corresponding time step.

Table 4-2 Procedure of short-term residential load forecasting based on deep learning and k-means clustering

K-means Clustering	1) Select and preprocess residential load data to generate load profiles 2) Set a value for K 3) Perform k-means clustering T times, and according to the optimal clustering result, separate the load profiles into K clusters
Residential Load Forecasting	4) Generate a sample pool for a specific cluster and standardize every sample 5) Set values for the hyperparameters for an LSTM network 6) Train the LSTM network on the sample pool 7) If the LSTM networks of all the clusters are obtained, go to 8); otherwise, repeat 4)-6) for another cluster 8) Evaluate the prediction performance for each residential customer

In order to learn both similarity and distinction of residential load effectively, k-means clustering is firstly employed to divide residential load profiles into K clusters for the purpose of generating a sample pool for each cluster, and the sample pool is comprised of the load profiles of the residents of the same cluster. Then, an LSTM network is trained on this sample pool and used to predict the load of every resident in this cluster.

As a result, the procedure of short-term residential load forecasting based on deep learning and k-means clustering is shown in Table 4-2.

4.5.2 Dataset Description

The dataset used in this chapter is from the Smart Metering Electricity Customer Behaviour Trials initiated by the Commission for Energy Regulation in Ireland [140]. The trials lasted from 01/07/2009 to 31/12/2010 with over 5000 Irish residential customers and small and medium enterprises participating. The full dataset is publicly available online, and consists of three main parts: 1) half-hourly sampled electricity consumption from each participant; 2) questionnaires and corresponding answers of surveys; 3) customer type and tariff and stimulus description.

It is noted that there are different levels of missing intervals for a number of participants in this dataset. Hence, data preprocessing is very necessary for these participants before using their data.

4.5.3 Experiment Setup

It is a common practice for machine learning that a full dataset is divided into a training dataset and a test dataset with a ratio of 9:1. So, as for each residential customer, 90% of the data samples are used for training, while the rest of 10% are used for testing. In addition, as this chapter is not focused on improving the prediction accuracy via the optimal network structure, hyperparameter fine-tuning is not conducted on the LSTM network here. All the experiment parameters are presented in Table 4-3.

Table 4-3 Experiment parameters of short-term residential load forecasting based on deep learning and k-means clustering

RNN layer number	5
Fully-connected layer number	1
Neuron number of RNN layer	128
Neuron number of fully-connected layer	128
Batch size	128
Time length of input	24

RNN unit	LSTM
Activation function of RNN layer	tanh
Activation function of fully-connected layer	linear
Optimization method	AdamOptimizer
Training epoch	15
Learning rate	0.001
Loss function	RMSE

4.6 Case Studies and Results

In this section, a performance comparison is conducted between the proposed method in this chapter and the PDRNN developed in reference [137]. RMSE and MAE are utilised as the performance indexes for residential load forecasting. After that, the effect of k-means clustering on the forecasting performance of the proposed method is investigated. Besides, 50 residents are selected randomly from the Irish dataset to verify the proposed method, and their IDs are shown in Table 4-4. In order to make an equal performance comparison, the parameter *K* of k-means clustering is set as 5, because the PDRNN divides the 50 residents equally into 5 groups each containing 10 residents.

Table 4-4 IDs of the randomly selected residents

No.	ID	No.	ID	No.	ID	No.	ID	No.	ID
1	1002	11	1316	21	1720	31	2529	41	3262
2	1022	12	1440	22	1727	32	2732	42	3306
3	1027	13	1492	23	1797	33	2789	43	3970
4	1030	14	1494	24	1807	34	2793	44	4092
5	1076	15	1507	25	1808	35	2811	45	4279
6	1120	16	1530	26	1874	36	2967	46	4441
7	1229	17	1532	27	1883	37	2968	47	4626
8	1277	18	1624	28	2023	38	2971	48	5019
9	1281	19	1660	29	2106	39	3077	49	5237
10	1301	20	1670	30	2463	40	3080	50	5616

4.6.1 Performance Comparison of Short-Term Residential Load Forecasting

A performace comparison is made between the proposed method and the PDRNN in terms of forecasting accuracy. As different initializations for k-means clustering may lead to different clustering results, a reasonable way to obtain a comparetively optimal clustering result is that k-means clustering is run on the experimental dataset a number of times and then the best clustering result with the lowest loss is selected. In this chapter, k-means clustering is run $T = 25$ times on the 50 residents in order to obtain the optimal clustering result. In addition, as for the PDRNN, the 50 residents are randomly partitioned into 5 groups each containing 10 residents. The results of k-means clustering and random partitioning are presented in Table 4-5 and Table 4-6 respectively. Furthermore, the forecasting results of the proposed method and the PDRNN are shown in Table 4-7 and Table 4-8 respectively.

Table 4-5 Results of k-means clustering of the proposed method

Cluster No.	Number of Residents	IDs
1	9	1229, 1660, 2789, 3077, 3080, 3306, 3970, 4441, 4626
2	21	1002, 1022, 1027, 1030, 1120, 1277, 1316, 1440, 1492, 1507, 1530, 1797, 1807, 1808, 1874, 1883, 2106, 4092, 4279, 5237, 5616
3	3	2793, 2811, 2971
4	5	1494, 2732, 2968, 3262, 5019
5	12	1076, 1281, 1301, 1532, 1624, 1670, 1720, 1727, 2023, 2463, 2529, 2967

Table 4-6 Results of random partitioning of the PDRNN

Group No.	Number of Residents	IDs
1	10	1120, 1281, 1492, 1530, 2106, 2811, 2967, 3970, 4441, 5019

2	10	1022, 1277, 1301, 1720, 1797, 1808, 1874, 2529, 2793, 2968
3	10	1316, 1507, 1532, 1727, 1807, 1883, 2463, 3080, 3306, 5616
4	10	1027, 1229, 1440, 1660, 1670, 2732, 3077, 3262, 4626, 5237
5	10	1002, 1030, 1076, 1494, 1624, 2023, 2789, 2971, 4092, 4279

Table 4-7 Forecasting results of the proposed method

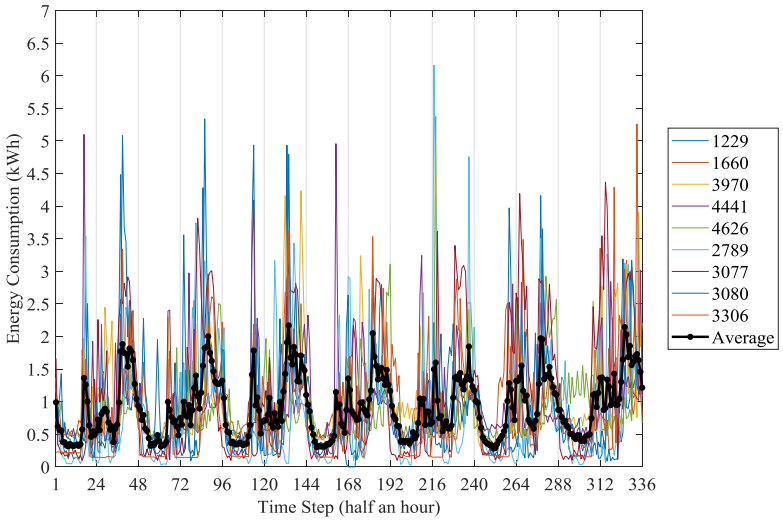
Cluster No.	RMSE (kWh)	MAE (kWh)
1	0.547	0.339
2	0.279	0.16
3	0.601	0.327
4	0.556	0.329
5	0.401	0.25
All Residents	**0.404**	**0.241**

Table 4-8 Forecasting results of the PDRNN

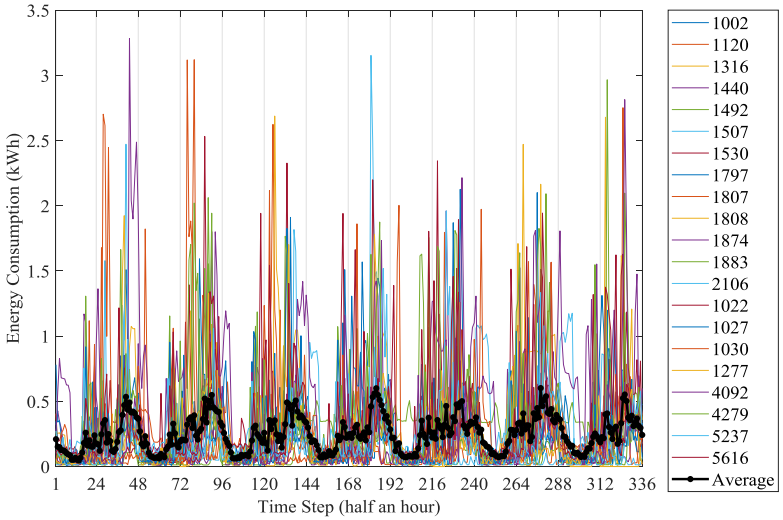
Group No.	RMSE (kWh)	MAE (kWh)
1	0.522	0.394
2	0.459	0.294
3	0.449	0.325
4	0.531	0.372
5	0.488	0.328
All Residents	**0.49**	**0.343**

It is noted that Table 4-7 describes the average RMSEs and MAEs of each cluster and all residents, while Table 4-8 describes the average RMSEs and MAEs of each group and all residents. In terms of the average RMSE of all residents, it can be easily seen from Table 4-7 and Table 4-8 that the proposed method performs residential load forecasting much better than the PDRNN, having a dramatic decrease of $\frac{0.49-0.404}{0.49} \times$

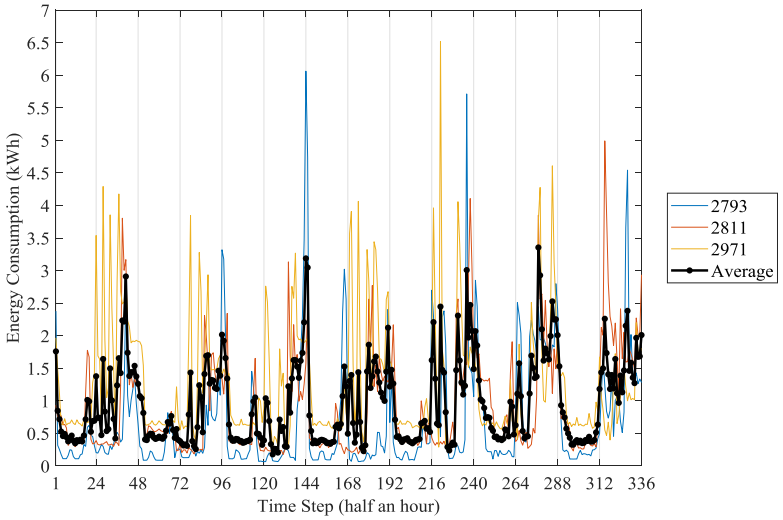
$100\% = 17.55\%$. Likewise, compared with the PDRNN, the proposed method decreases the average MAE of all residents significantly by $\frac{0.343-0.241}{0.343} \times 100\% = 29.74\%$. The reason for the performance improvement is explained as follows. As k-means clustering divides the residents into clusters based on their similarity, it is comparetively easier for the proposed method to learn the common energy comsumption characteristics within a particular cluster, meanwhile capturing the distinctive features of each single resident. However, the PDRNN simply seperates the residents into groups in a random way, very likely to incur the significant energy consumption differences within a single group. This largely impedes the PDRNN from extracting the energy consumption similarity of each group. As a result, the proposed method achieves a much higher prediction accuracy than the PDRNN.



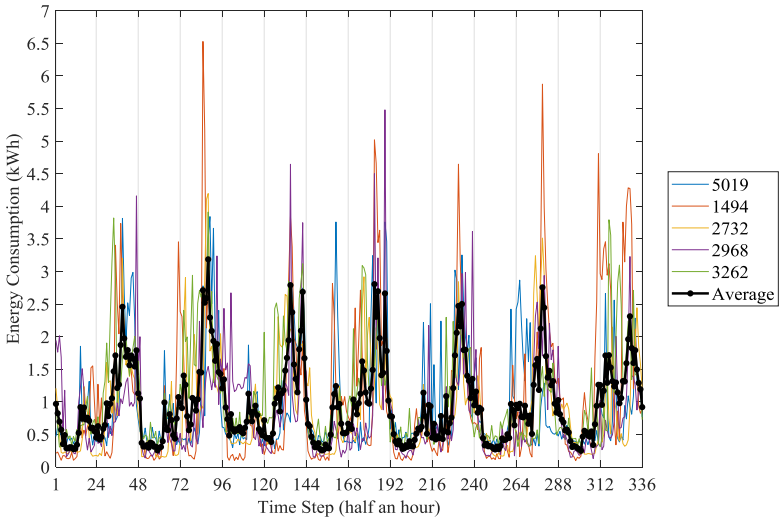
(a) Cluster 1



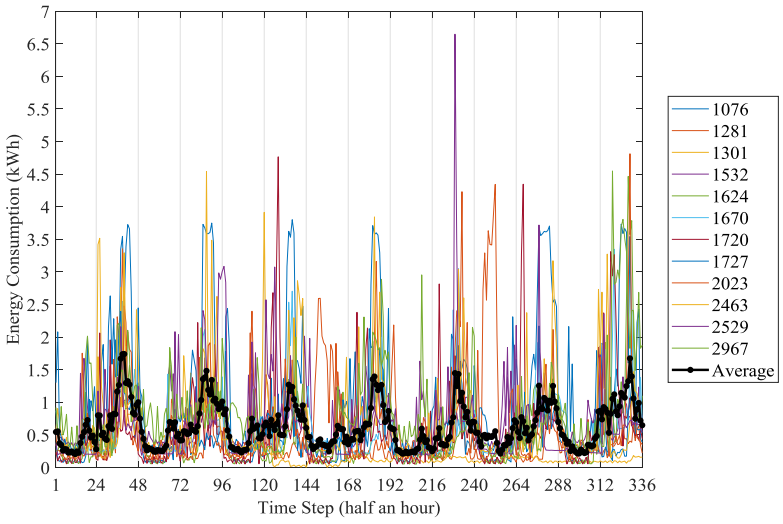
(b) Cluster 2



(c) Cluster 3

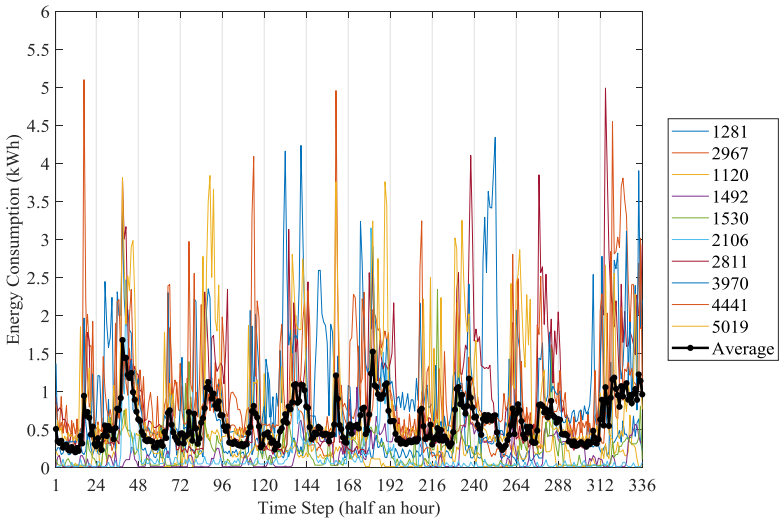


(d) Cluster 4

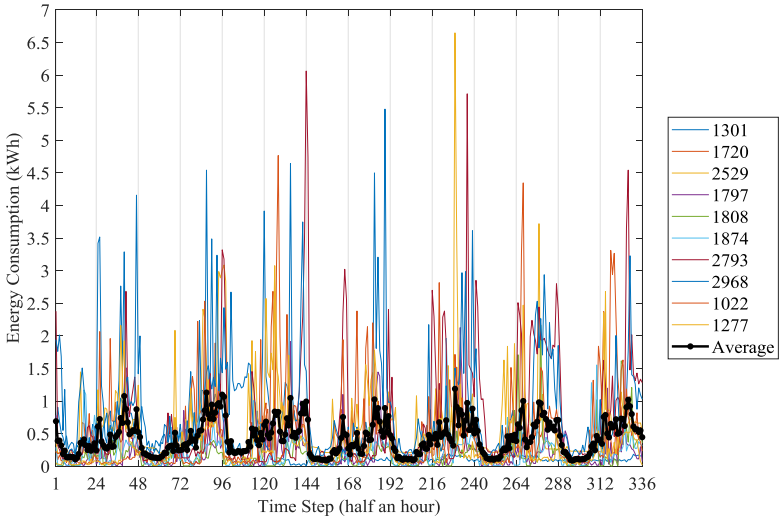


(e) Cluster 5

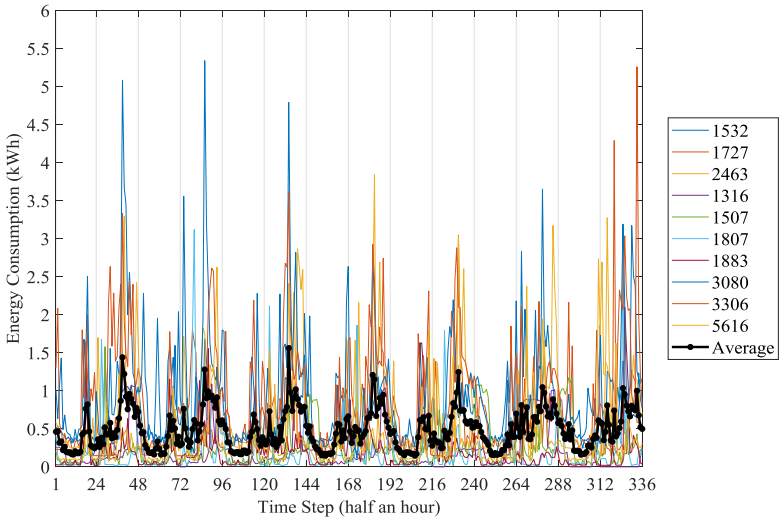
Figure 4-4 Load profiles of all the clusters of the proposed method



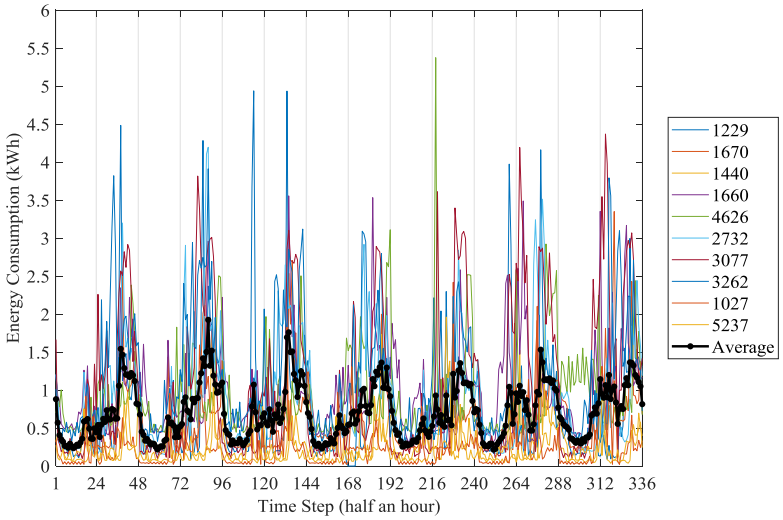
(a) Group 1



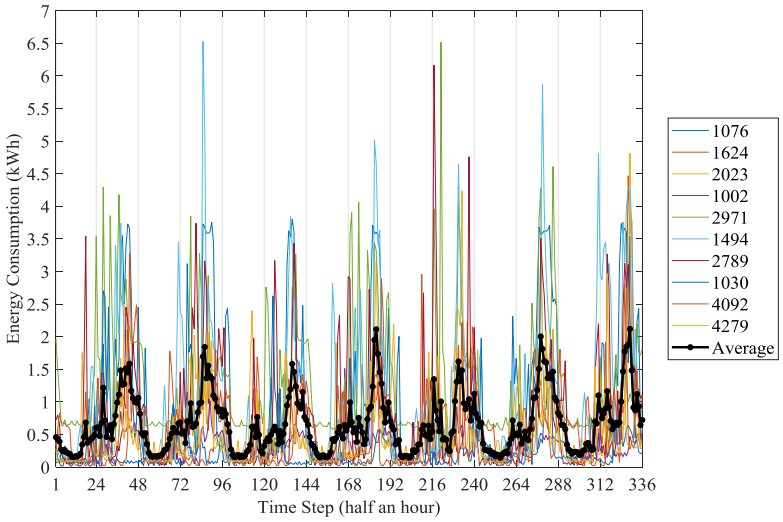
(b) Group 2



(c) Group 3



(d) Group 4



(e) Group 5

Figure 4-5 Load profiles of all the groups of the PDRNN

Furthermore, Figure 4-4 shows the load profiles of the residents of each cluster of the proposed method from 18/01/2010 (Monday) to 24/01/2010 (Sunday), while Figure 4-5 shows the load profiles of the residents of each group of the PDRNN over the same period. The bold black lines in both figures represent the average load profiles of the clusters and groups.

According to Figure 4-4, it can be easily discovered that the average load profile of each cluster can represent the common energy consumption pattern of each cluster to a large extent. Besides, the average load profile of each cluster has a moderate pattern repetition during weekdays and weekends respectively. This is exactly in accordance

with a commonly accepted fact that household residents have different energy consumption behaviours over weekdays and weekends respectively. However, in Figure 4-5, it is obviously seen that the average load profile of each group is unable to represent the residential load profiles of each group as a whole. Also, there is no pattern repetition in the average load profile of each group during both weekdays and weekends. The reason for it is elaborated as follows. As the PDRNN partitions the residents into groups at random, different energy consumption patterns are very likely to appear in the same group. Thus, a similar energy consumption pattern cannot be extracted from their load profiles. But, the proposed method employs k-means clustering to divide the residents into clusters on the basis of their similarity. As a result, the inherent common energy consumption characteristics of each cluster can be easily reflected, enabling the proposed method to achieve a much better forecasting accuracy.

4.6.2 Effect of K-means Clustering on Forecasting Performance

As different initialization results tend to lead to different clustering results, the effect of k-means clustering on the forecasting performance of the proposed method is further investigated. Table 4-9 presents the forecasting results of the proposed method when k-means clustering is run for the tenth time, while Table 4-10 presents the forecasting results of the proposed method when k-means clustering is run for the fifteenth time. In these two cases, the overall losses of k-means clustering are both less than the corresponding overall loss of k-means clustering in Table 4-7.

Table 4-9 Forecasting results of the proposed method
(k-means clustering is run for the tenth time)

Cluster No.	Number	RMSE (kWh)	MAE (kWh)
1	1	0.652	0.376
2	11	0.412	0.255
3	8	0.583	0.334
4	8	0.554	0.344
5	22	0.3	0.172
All Residents	50	**0.418**	**0.248**

Table 4-10 Forecasting results of the proposed method (k-means clustering is run for the fifteenth time)

Cluster No.	Number	RMSE (kWh)	MAE (kWh)
1	1	0.591	0.39
2	1	0.65	0.381
3	22	0.491	0.316
4	21	0.28	0.159
5	5	0.559	0.342
All Residents	50	**0.414**	**0.255**

It can be obviously observed that the average RMSEs and MAEs of all residents in Table 4-9 and Table 4-10 are both slightly higher than the corresponding indexes in Table 4-7. In addition, it is noted that cluster 1 only contains one resident in Table 4-9, while cluster 1 and cluster 2 both contain only one resident in Table 4-10. To be specific, whether k-means clustering is run for the tenth or fifteenth time, the clustering results both have the clusters with only one resident. This makes the other clusters have excessive residents and then results in the unobvious similarity for them. Consequently, the LSTM models are only able to learn the less similarity from them, finally leading to a slightly worse forecasting accuracy, regardless of RMSE or MAE. However, it can be also observed obviously that the average RMSEs and MAEs of all residents in Table 4-9 and Table 4-10 are both much lower than the corresponding indexes in Table 4-8. This is because k-means clustering can still separate the residents more reasonably than random partitioning and extract the moderate similarity from them, even if one or two clusters have only one resident and several others have excessive residents. So, when selecting the clustering result after running k-means clustering a number of times, the cluster distribution should be carefully taken into account as well, although the overall loss of clustering plays a crucial role in the selection for the clustering result.

4.7 Chapter Summary

This chapter developed a short-term individual residential load forecasting method based on deep learning and k-means clustering. Besides, a comprehensive feature expression strategy is used to provide the developed forecasting model with sufficient information at each time step in order to increase the forecasting accuracy. The

experimental results have shown that the developed short-term residential load forecasting method has improved the forecasting accuracy significantly in terms of RMSE and MAE, compared with the PDRNN. Moreover, the effect of k-means clustering on the developed method is further explored, and the comparison results have suggested that both the cluster distribution and overall loss of k-means clustering should be taken into consideration for the purpose of improving the forecasting accuracy, when combining the clustering result with the LSTM models.

Chapter 5

Online Residential Load Forecasting Based on Deep Learning and Dynamic Mirror Descent

5.1 Introduction

As advanced metering infrastructure (AMI) is being widely deployed in the modern power system, especially, smart meters, a growing number of granular data of residential electricity consumption has become easily available on a large scale [141]. This huge amount of data has enabled power network operators to motivate residential users to actively participate in demand side management (DSM) through a wide range of various demand response programs (DRPs), for example, time-of-use pricing [142-143]. As part of DSM, residential load forecasting is a significantly important but challenging task for power network operators, due to great irregularity and uncertainty of residential load. Consequently, addressing the challenge of residential load forecasting plays a crucial role in DSM.

Currently, residential load forecasting is generally categorised as aggregate-level and individual-level. More specifically, aggregate-level residential load forecasting methods mainly include support vector regression (SVR) [144-145], artificial neural networks (ANNs) [139, 146-149], and deep neural networks (DNNs) [150]. Besides, these methods tend to be combined with clustering techniques (e.g. k-means clustering) by many researchers for the purpose of improving the forecasting performance. In general, a series of different models based on these methods have obtained a desirable prediction accuracy for aggregate-level residential load forecasting. This is because a number of diverse behaviours of residential customers can smooth out the overall load profile of those residential customers at the aggregate level, therefore generating an easily identifiable pattern.

However, compared to aggregate-level residential load forecasting, only a few researchers have so far attempted to explore individual-level residential load forecasting. Some traditional forecasting methods, for instance, ANNs, are still applied to forecast individual residential load [151-152]. During most recent years, DNNs have been largely adopted due to their superior capability to extract complex patterns, such as LSTM networks and RNNs [153-156]. Although the DNN models have mostly achieved a higher forecasting accuracy than the traditional models, they are still basically trained offline and then applied online. In other words, if the training data is not selected properly or sufficiently to train a DNN forecasting model and then the

obtained model is directly applied online, it is very likely to have a severely negative effect on its forecasting performance, because residential load tends to change significantly over time.

Hence, different from Chapter 4, this chapter proposes an online individual residential load forecasting method based on an effective combination of deep learning and dynamic mirror descent (DMD), which is able to forecast residential load in real time and adjust the forecasting error over time in order to improve the forecasting performance. To be specific, it firstly employs an LSTM network to build a prediction model, and then applies it online with DMD correcting the prediction error. In order to increase the prediction accuracy, a comprehensive feature expression strategy is utilised to describe load characteristics of each time step in detail. Finally, the proposed online residential load forecasting method is evaluated on a real-life Irish residential load dataset. The experimental results indicate that it can obtain a higher prediction accuracy in terms of both RMSE and MAE.

5.2 Overview of Residential Load Forecasting

A number of research works have been presented in the area of aggregate-level residential load forecasting. Reference [144] designed a short-term cluster-based residential load forecasting strategy for the aggregate level. It firstly clusters residential customers, then forecasts the energy consumption of each cluster separately through SVR, and finally aggregates the energy consumption forecasts of all the clusters. Similar to reference [144], reference [145] developed a residential load forecasting method for the district level, which combines k-means clustering with SVR.

Besides, ANNs have been commonly employed to forecast residential load at the aggregate level. For example, a new dynamic forecasting mechanism was proposed to actively monitor small-scale residential electricity demand and detect anomalous pattern changes in reference [146]. A self-organizing map is employed for anomalous day detection, and an ANN prediction model changes its input neurons according to a previously detected and recorded match in a database of anomalous days in order to conduct demand prediction for anomalous days. Franklin presented a three-step aggregate-level residential load forecasting approach, based on k-means clustering and

ANNs [139]. Firstly, this approach applies k-means clustering to group residential customers according to similarity in energy consumption behaviours. Then, an ANN model is used to conduct load forecasting for each group. Finally, the forecast of each group is summed to obtain the total residential load forecast. Reference [148] and reference [149] both made a comprehensive performance comparison between ANNs and some other prediction models, such as auto-regression and auto-regressive integrated moving average. Moreover, a GRU neural network based approach was developed to perform short-term load forecasting for residential community [150]. Also, it uses LASSO and partial correlation analysis to explore the influences of temperature, humidity, rainfall, and wind speed on residential load in order to determine the input variables for the forecasting model. In summary, aggregate residential load is comparatively easier to forecast than individual residential load, because a large number of different residential behaviours can cancel out the volatility and uncertainty of residential energy consumption to a large extent. As a result, existing forecasting models for aggregate residential load have obtained a satisfactory prediction accuracy.

However, as for individual-level residential load forecasting, only a few efforts have been made by researchers during recent years. Xu proposed a novel k-nearest vector auto-regressive framework with exogenous input to spatial-temporally model household-level electricity demand [151]. In addition, Dinesh presented a forecasting approach to the power consumption of a single household, which is based on non-intrusive load monitoring (NILM) and graph spectral clustering [152]. Within the approach, the aggregate power signal is decomposed into individual appliance signals, and then the power consumption of each appliance is forecasted respectively. Finally, the total power forecast of a single household is formed by aggregating the power forecasts of all the appliances. Different from reference [151] and reference [152], reference [153] developed a probabilistic forecasting model in order to describe the uncertainty of individual residential load using density-estimating ANNs. Two different types of probabilistic ANNs are implemented and compared, which are mixture density networks and softmax regression networks respectively. Similarly, reference [154] employed a quantile LSTM network to perform probabilistic residential load forecasting. Furthermore, different DNNs, including RNNs, LSTM networks, and GRU networks, were applied to short-term residential load forecasting at the individual level and a performance comparison was conducted among them [155]. In reference [156],

automatic hyperparameter tuning was utilised to select an optimal hyperparameter combination for an LSTM network in order to improve the accuracy of individual residential load forecasting. Although a variety of forecasting models have been developed, these offline well-trained models are still quite likely to encounter many sudden changes, which are not seen during training, when applied online, because of high volatility and uncertainty of individual residential load. Considering this issue and very few efforts made on online residential load forecasting at the individual level, this chapter presents solutions to it.

The remainder of this chapter is organized as follows. Section 5.3 introduces dynamic mirror descent in detail. Section 5.4 formulates the effective combination of deep learning and dynamic mirror descent for online residential load forecasting. Section 5.5 evaluates the proposed online residential load forecasting method on a real-world Irish residential load dataset. Finally, the conclusions are given in Section 5.6.

5.3 Dynamic Mirror Descent

As a novel online learning method, dynamic mirror descent (DMD) is capable to incorporate a dynamic model in the learning process, compared to classical online learning methods. It can effectively minimize the loss and estimate time-varying system states [157-158]. So, in this chapter, the original DMD is introduced briefly in the first place, and then a few modifications are made to enable the original DMD to become applicable to residential load forecasting.

Table 5-1 Learning process of DMD

1) Given a non-increasing sequence of step size $\eta_t > 0$, initialise $\hat{\theta}_1 \in \Theta$
2) Observe x_t and incur loss $l_t(\hat{\theta}_t)$ $(t = 1, 2, \cdots, T)$
3) Environment produces a dynamic model Φ_t
4) Calculate $\tilde{\theta}_{t+1} = \arg\min_{\theta \in \Theta} \langle \nabla f_t(\hat{\theta}_t), \theta \rangle + \eta_t r(\theta) + D(\theta \| \hat{\theta}_t)$
5) Calculate $\hat{\theta}_{t+1} = \Phi_t(\tilde{\theta}_{t+1})$
6) If $t < T$, repeat 2)-5) at the next time t+1; otherwise, end the learning process of DMD

DMD is executed by two main steps: 1) an observation-based update incorporates the new measurement into the parameter prediction; 2) a model-based update advances the parameter prediction to the next time step. The detailed learning process of DMD is presented in Table 5-1.

In Table 5-1, θ is the system state; Θ is a convex set; x is the observed data point; η_t is a step size parameter; Φ_t is a known dynamic model based on prior knowledge; $l_t(\cdot)$ is a convex loss function, and it is the composition of a convex function $f_t(\cdot)$ from the environment; $r(\cdot)$ is a convex regularization function and does not change over time; $\nabla f_t(\cdot)$ is an arbitrary subgradient function of $f_t(\cdot)$; $\hat{\theta}_t$ is the final prediction; $\langle \cdot\,, \cdot \rangle$ is the dot product operator; $D\left(\theta||\hat{\theta}_t\right)$ is the Bregman divergence measuring the distance between θ and $\hat{\theta}$.

In order to apply DMD to forecast residential load online, a few modifications are made to the original DMD. The idea is that the concept of the original DMD is still adopted but it is not a direct implementation of the original DMD. In other words, the modified DMD regards the prediction model as a black box and simply adjusts its output with the measured and forecasted values. More specifically, the modified DMD is formulated as follows [158]:

$$\hat{k}_{t+1} = \arg\min_{\theta\in\Theta} \eta\langle \nabla l_t\left(\hat{\theta}_t\right), \theta\rangle + D\left(\theta||\hat{k}_t\right) \tag{5-1}$$

$$\breve{\theta}_{t+1} = \Phi(\bar{\theta}_t) \tag{5-2}$$

$$\hat{\theta}_{t+1} = \breve{\theta}_{t+1} + \hat{k}_{t+1} \tag{5-3}$$

where ∇l_t is an arbitrary subgradient function of $l_t(\cdot)$; $\hat{k}_t$ is the adjustment variable accumulating the deviation between the forecasted and measured values; η is a constant step size; Φ is the residential load forecasting model; $\bar{\theta}_t$ is the input sample of Φ. The model-based update (5-2) computes an open-loop prediction $\breve{\theta}_{t+1}$, which means that the actual measurement does not influence $\breve{\theta}_{t+1}$. The measurement-based update and the model-based prediction are combined in (5-3). Apparently, the modified DMD uses a closed-loop model-based update process, where the convex function (5-1) adjusts the parameter estimate into $\hat{k}_{t+1}$, which is used to compute $\hat{\theta}_{t+1}$ together with $\breve{\theta}_{t+1}$.

In this chapter, for simplicity, the loss function $l_t(\hat{\theta}_t)$ is selected as $l_t(\hat{\theta}_t) = \frac{1}{2}\|\hat{\theta}_t - y_t\|_2^2$, while the Bregman divergence $D(\theta||\hat{k}_t)$ is selected as $D(\theta||\hat{k}_t) = \frac{1}{2}\|\theta - \hat{k}_t\|_2^2$. Thus, the convex function (5-1) can be simplified as [158]

$$\hat{k}_{t+1} = \hat{k}_t + \eta(y_t - \hat{\theta}_t) \tag{5-4}$$

where y_t is the actual measurement.

As a result, the modified DMD is formed by (5-2)-(5-4) in order for it to become applicable to online residential load forecasting. In the next section, the combination of the modified DMD and deep learning will be elaborated.

5.4 Methodology Formulation

Similar to Chapter 4, this chapter still employs LSTM networks to build prediction models. However, as they have already been introduced in detail in Chapter 4, they are not included again in this chapter. Thus, the implementation process of the proposed online residential load forecasting method and the experiment setup are formulated respectively below.

5.4.1 Implementation Process

Due to high volatility and uncertainty of individual residential load, a comprehensive feature description strategy, which is similar to the strategy used in Chapter 4, is exploited in this chapter in order to improve the forecasting accuracy. So, the input features of a data sample $\mathbf{S}_t$ at a particular time step t are detailed as follows [136]:

1) the sequence of the residential load for the past T time steps $\mathbf{E}_t = [e_{t-T}, \cdots, e_{t-2}, e_{t-1}] \in \mathbf{R}^T$;

2) the sequence of the time indexes for the past T time steps $\mathbf{D}_t \in \mathbf{R}^T$, where the range for each element is [1, 48], because the sampling frequency is once every half an hour;

3) the sequence of the day indexes for the past T time steps $\mathbf{W}_t \in \mathbf{R}^T$, where each element ranges from 1 to 7, as there are 7 days in a week;

4) the sequence of the holiday signs for the past T time steps $\mathbf{H}_t \in \mathbf{R}^T$, where each element is either 1 or 2, and 1 denotes non-holiday and 2 denotes holiday (in this

chapter, it is assumed that weekdays are non-holiday and weekends are holiday).

So, a data sample $\mathbf{S}_t = \left[\mathbf{E}_t^{\mathrm{T}}, \mathbf{D}_t^{\mathrm{T}}, \mathbf{W}_t^{\mathrm{T}}, \mathbf{H}_t^{\mathrm{T}}\right]$ is a matrix of a concatenation of the four sequences. In order to speed up the convergence of the LSTM network and improve its generalization capacity, the input features are standardized to [0, 1] according to their nature. To be specific, the min-max standardization method is adopted for $\mathbf{E}_t$, while $\mathbf{D}_t$, $\mathbf{W}_t$, and $\mathbf{H}_t$ are encoded by a one-hot encoder. The one-hot encoder maps an original element of the feature sequence with M categories into a new sequence with M elements, where only the new element corresponding to the original element is one while the rest are all zero. Hence, a standardized data sample $\widehat{\mathbf{S}}_t = \left[\widehat{\mathbf{E}_t^{\mathrm{T}}}, \widehat{\mathbf{D}_t^{\mathrm{T}}}, \widehat{\mathbf{W}_t^{\mathrm{T}}}, \widehat{\mathbf{H}_t^{\mathrm{T}}}\right]$ is a $T \times N$ matrix, where $N = 1 + 48 + 7 + 2 = 58$. Each row of the standardized input sample $\widehat{\mathbf{S}}_t$ is the detailed features for the corresponding time step.

Table 5-2 Procedure of online residential load forecasting based on deep learning and dynamic mirror descent

Offline Training	1) Select and preprocess residential load data 2) Generate a training dataset and a test dataset, and standardize every sample 3) Set values for the hyperparameters for an LSTM network 4) Train the LSTM network on the training dataset
Online Forecasting with DMD	5) Initialise the step size η, the time length $T_{\max}$, and the adjustment variable $\hat{k}_t$ 6) Calculate $\breve{\theta}_t$ using (5-2), $t = 1, 2, \cdots, T_{\max}$ 7) Calculate $\hat{k}_t$ using (5-4) 8) Calculate $\hat{\theta}_t$ using (5-3) 9) If $t < T_{\max}$, repeat 6)-8) at the next time step t+1; otherwise, go to 10) 10) Evaluate the prediction performance on the test dataset

In order to perform online residential load forecasting, an LSTM network is well trained offline in the first place, and then it is applied online with DMD adjusting the forecast in real time. Hence, the procedure of online residential load forecasting based on deep learning and dynamic mirror descent is shown in Table 5-2.

5.4.2 Experiment Setup

The same residential load dataset from Ireland is used in this chapter as used in Chapter 4, and the full dataset of a single resident is divided into a training dataset and a test dataset with a ratio of 9:1. So, for each resident, 90% of the data samples are used for training, while the rest of 10% are used for testing. In addition, as this chapter is not focused on improving the prediction accuracy via the optimal network structure, hyperparameter fine-tuning is not conducted on the LSTM network here. All the experiment parameters are presented in Table 5-3.

Table 5-3 Experiment parameters of online residential load forecasting based on deep learning and dynamic mirror descent

RNN layer number	3
Fully-connected layer number	1
Neuron number of RNN layer	64
Neuron number of fully-connected layer	64
Batch size	128
Time length of input	12
RNN unit	LSTM
Activation function of RNN layer	tanh
Activation function of fully-connected layer	linear
Optimization method	AdamOptimizer
Training epoch	50
Learning rate	0.001
Loss function	RMSE

5.5 Case Studies and Results

In this section, a performance comparison is made between offline and online individual residential load forecasting. It is noted here that offline means forecasting without online learning, while online means forecasting with online learning. RMSE and MAE are employed as the performance indexes for residential load forecasting. Afterwards, the effect of the parameter η of the modified DMD on online residential load forecasting is further investigated. In addition, 30 residents are selected at random from the Irish dataset to validate the proposed method, and their IDs are shown in Table 5-4. The adjustment variable $\hat{k}_1$ is initialised as 0.

Table 5-4 IDs of the randomly selected residents

No.	ID	No.	ID	No.	ID
1	1076	11	2106	21	2971
2	1277	12	2424	22	3262
3	1281	13	2485	23	3314
4	1301	14	2490	24	3337
5	1440	15	2529	25	3536
6	1492	16	2547	26	4176
7	1507	17	2732	27	4441
8	1607	18	2811	28	5019
9	1670	19	2918	29	5551
10	2055	20	2967	30	6240

5.5.1 Performance Comparison of Offline and Online Residential Load Forecasting

A performace comparison is performed between offline and online residential load forecasting in terms of prediction accuracy. In this case, the parameter η of the modified DMD is set as 1.0×10^{-2}. The results of offline and online residential load forecasting are shown in Table 5-5.

Table 5-5 Performance comparison between offline and online residential load forecasting

No.	MAE			RMSE		
	offline (kWh)	online (kWh)	improvement (%)	offline (kWh)	online (kWh)	improvement (%)
1	0.2	0.2	0	0.36	0.36	0
2	0.21	0.2	4.8	0.4	0.37	7.5
3	0.24	0.2	16.7	0.38	0.37	2.6
4	0.28	0.23	17.9	0.44	0.43	2.3
5	0.12	0.1	16.7	0.17	0.16	5.9
6	0.2	0.11	45	0.28	0.17	39.3
7	0.16	0.12	25	0.25	0.19	24
8	0.06	0.02	66.7	0.1	0.04	60
9	0.27	0.21	22.2	0.43	0.36	16.3
10	0.21	0.16	23.8	0.39	0.28	28.2
11	0.14	0.1	28.6	0.32	0.31	3.2
12	0.07	0.05	28.6	0.09	0.08	11.1
13	0.29	0.27	6.9	0.48	0.47	2.1
14	0.21	0.16	23.8	0.36	0.32	11.1
15	0.24	0.22	8.3	0.41	0.4	2.4
16	0.31	0.29	6.5	0.52	0.48	7.7
17	0.35	0.34	2.9	0.6	0.53	11.7
18	0.29	0.31	-6.9	0.57	0.54	5.3
19	0.1	0.08	20	0.16	0.15	6.3
20	0.26	0.26	0	0.39	0.39	0
21	0.33	0.34	-3	0.62	0.61	1.6
22	0.3	0.28	6.7	0.51	0.45	11.8
23	0.2	0.2	0	0.36	0.36	0
24	0.27	0.23	14.8	0.39	0.38	2.6
25	0.31	0.29	6.5	0.52	0.46	11.5
26	0.18	0.16	11.1	0.44	0.45	-2.3
27	0.38	0.34	10.5	0.6	0.55	8.3

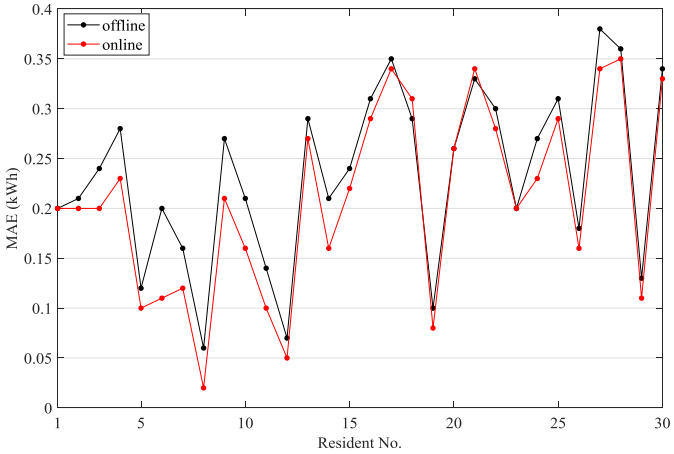
28	0.36	0.35	2.8	0.58	0.59	-1.7
29	0.13	0.11	15.4	0.26	0.25	3.8
30	0.34	0.33	2.9	0.53	0.51	3.8

In Table 5-5, it can be easily discovered that the proposed online method performs much better than the offline method for most residents, in terms of both MAE and RMSE. To be specific, the greatest improvement of MAE of the proposed online method is 66.7% (ID 1607), compared to the offline method. In contrast, the smallest improvement of MAE of the proposed online method is 2.8% (ID 5019). Likewise, the greatest improvement of RMSE of the proposed online method is 60% (ID 1607), while the smallest improvement of RMSE of the proposed online method is 1.6% (ID 2971). In summary, the online learning method, the modified DMD, is generally capable to effectively adjust the prediction error in real time, therefore improving the prediction accuracy to a different extent.

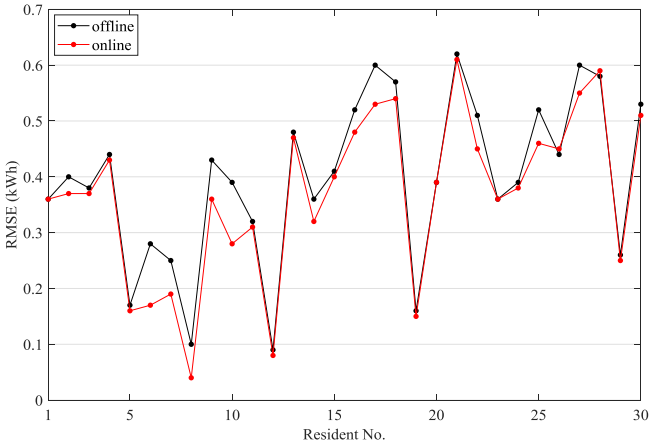
But, there are also a couple of special cases, such as ID 2811 and ID 4176. In these cases, one of the two performance indexes of the proposed online method is lower than that of the offline method, while the other one of the two performance indexes of the proposed online method is higher than that of the offline method. This is because MAE and RMSE indicate forecasting performance from two different perspectives. In other words, MAE, which reflects the mean of errors, regards every error equally and averages all the errors, while RMSE, which reflects the fluctuation of errors, strengthens the large error and weakens the small error.

Another interesting finding is that there is no performance difference between the proposed online method and the offline method for several residents, for example, ID 3314. This suggests that the modified DMD fails to effectively adjust the forecast over time, mainly because of the great complexity of these residential load profiles.

Furthermore, Figure 5-1 compares the MAEs and RMSEs of all residents between offline and online residential load forecasting, and Figure 5-2 depicts the actual and forecasted load profiles of a certain resident on 19/12/2010.



(a) MAEs of all residents



(b) RMSEs of all residents

Figure 5-1 Performance comparison between offline and online residential load forecasting

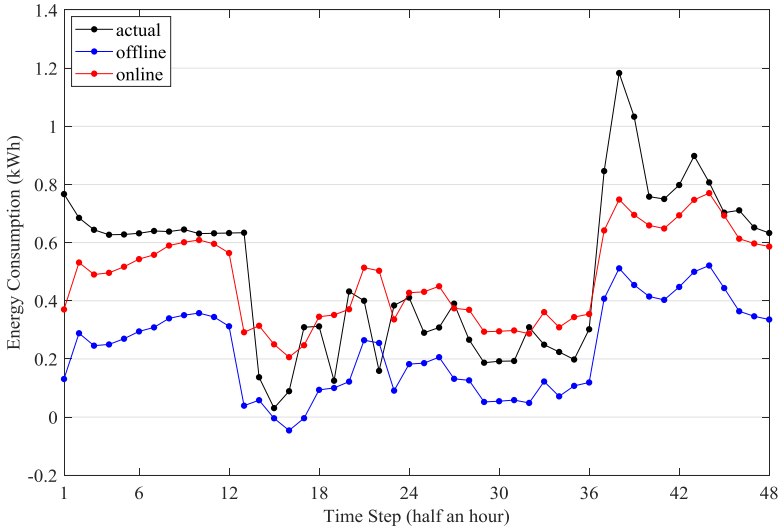


Figure 5-2 Load profiles of ID 1492

According to Figure 5-1, it can be obviously seen that the proposed online method tends to adjust the forecast more effectively when the forecasting error is smaller, regardless of MAE or RMSE. In other words, when the MAE or RMSE is comparatively smaller, the accuracy improvement of the proposed online method is much greater, such as No. 6 and No. 8. However, when the MAE or RMSE is comparatively larger, the accuracy improvement of the proposed online method is much smaller, such as No. 16 and No. 30. The reason for this is that if the residential load pattern is easier to extract, the modified DMD is accordingly able to track the error and adjust the forecast more accurately.

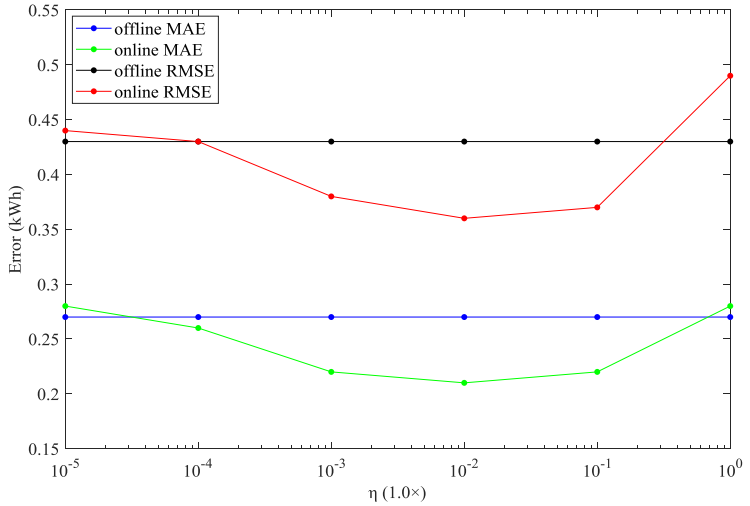
In Figure 5-2, the actual and forecasted load profiles of ID 1492 are taken as an example to demonstrate the superiority of the proposed online method. From time step 1 to time step 13 and from time step 37 to time step 48, the proposed online method forecasts much more precisely than the offline method, due to online learning. This indicates that the proposed online method outperforms the offline method substantially during the early morning and late evening. However, during the daytime of time step 14 to time step 36, the proposed online method performs similarly to the offline method.

5.5.2 Effect of Parameter η on Online Residential Load Forecasting

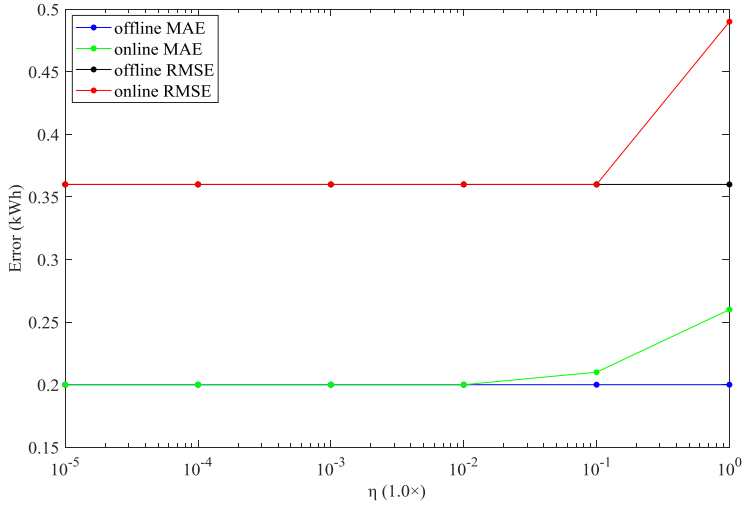
As the parameter η plays a rather important role in the modified DMD, its effect on the performance of online residential load forecasting is further investigated here. Three residents are selected to demonstrate the effect of the parameter η. Figure 5-3 illustrates the MAEs and RMSEs of the proposed online method when the modified DMD is applied with different values for the parameter η, and Figure 5-4 describes the forecasted load profiles of the proposed online method on 02/12/2010 when the modified DMD is applied with different values for the parameter η.

In Figure 5-3(a), it can be clearly seen that the proposed online method performs best for ID 1670 when η is 1.0×10^{-2}, in terms of both MAE and RMSE. This is because η effectively weighs the error between the actual and forecasted values, enabling the modified DMD to adjust the forecast accurately, which can be easily inferred from formula (5-4). Correspondently, in Figure 5-4(a), when η is 1.0×10^{-2}, the proposed

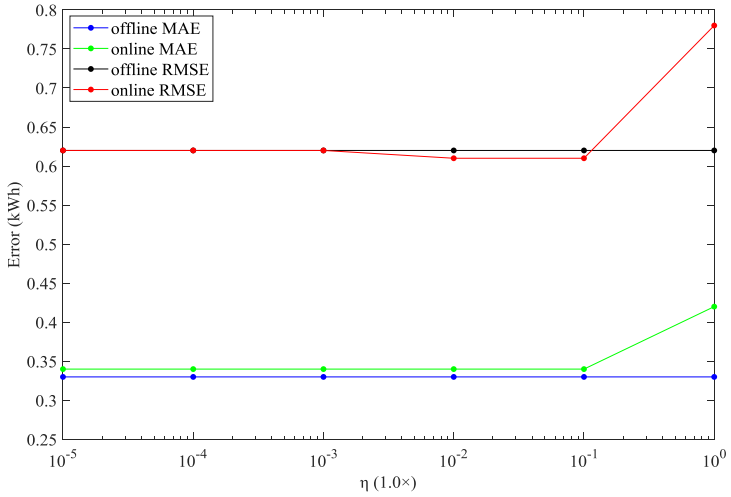
online method is capable to substantially decrease the error between the actual and forecasted values. But, it can be also observed in Figure 5-3(a) that the proposed online method performs even worse than the offline method when η is 1.0×10^{0} or 1.0×10^{-5}, in terms of both MAE and RMSE. This is because η fails to properly weigh the error between the actual and forecasted values, when it is too large or small. Therefore, the modified DMD is unable to track the actual value precisely. Besides, in Figure 5-4(a), it is obvious that the forecast of the proposed online method is far away from the actual value, when η is 1.0×10^{0} or 1.0×10^{-5}. All in all, when the modified DMD is applied in real time with the forecasting model, it is very possible to find out an optimal value for the parameter η, accordingly obtaining the maximum accuracy improvement.



(a) ID 1670

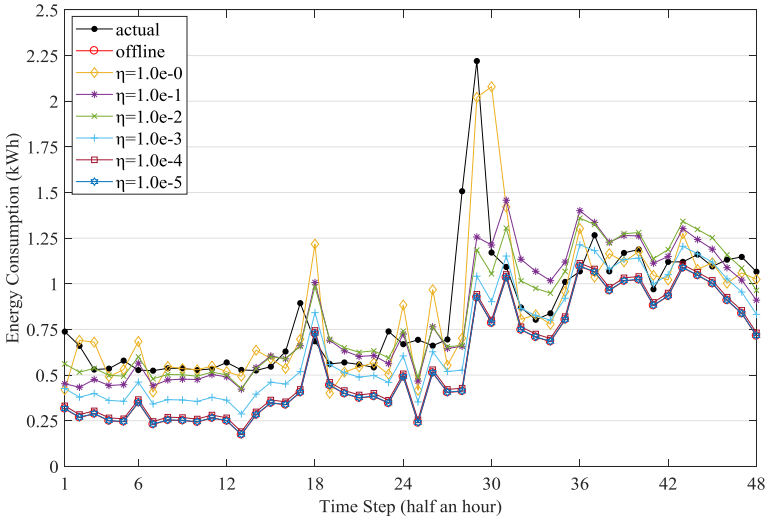


(b) ID 1076

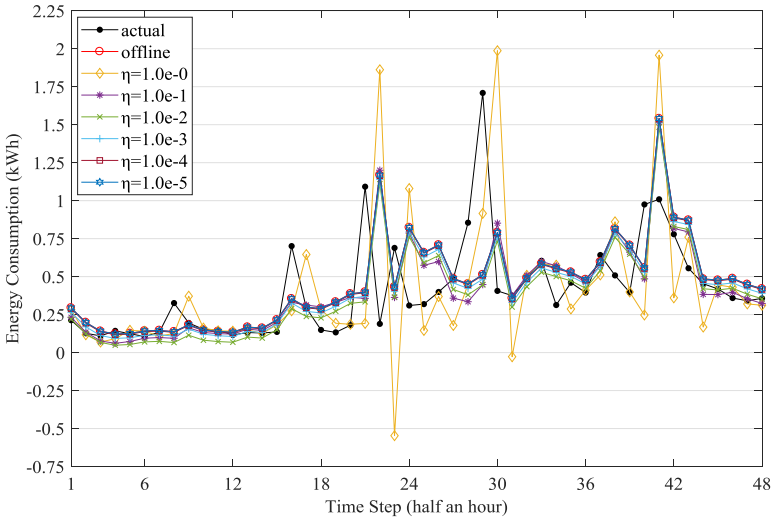


(c) ID 2971

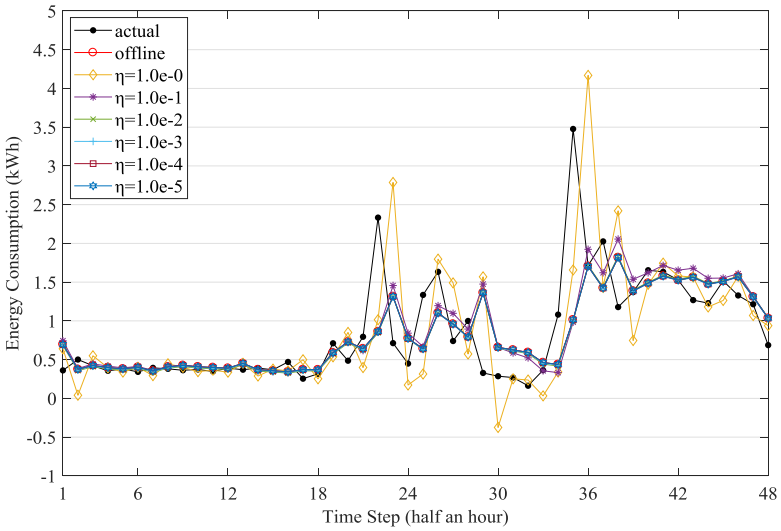
Figure 5-3 Effect of parameter η on online residential load forecasting



(a) ID 1670



(b) ID 1076



(c) ID 2971

Figure 5-4 Load profiles of online residential load forecasting for different values of parameter η

However, in Figure 5-3(b) and Figure 5-3(c), it can be easily discovered that the forecasting accuracy of the proposed online method is almost the same as that of the offline method, except that the proposed online method performs worse than the offline method when η is 1.0×10^0. The reason for that is briefed as follows. Due to the great complexity of the residential load profiles of ID 1076 and ID 2971, when η is much smaller than 1.0×10^0, the error between the actual and forecasted values is trusted much less while calculating the adjustment variable $\hat{k}_t$, according to formula (5-4). In other words, the adjustment variable $\hat{k}_t$ is calculated as a very small value at all time steps, making the modified DMD fail to track the actual value. As a consequence, the forecasted load profile of the proposed online method is quite similar to that of the offline method. Furthermore, as Figure 5-4(b) and Figure 5-4(c) show, the forecast of the proposed online method is very close to that of the offline method, when η is much smaller than 1.0×10^0. In general, although the modified DMD fails to track the actual value with a comparatively smaller value of the parameter η, it at least does not decrease the forecasting accuracy of the proposed online method, compared to the offline method.

5.6 Chapter Summary

This chapter proposed an online individual residential load forecasting method, based

on deep learning and dynamic mirror descent. A few modifications were made to the original DMD in order to apply it to online residential load forecasting. Besides, a comprehensive feature expression strategy is exploited to provide the proposed forecasting model with sufficient information at each time step for the purpose of increasing the forecasting accuracy. The experimental results have shown that the proposed online residential load forecasting method has improved the forecasting accuracy substantially in terms of both RMSE and MAE, compared to the offline method. In addition, the effect of the parameter η of the modified DMD on the proposed online method is further explored, and the comparison results have indicated that an optimal value of the parameter η is quite likely to be found out to achieve the maximum forecasting accuracy improvement, when applying the modified DMD to online residential load forecasting.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this section, the conclusions are drawn by summarising the major contributions and key findings of the thesis. There are three main works in the thesis, which are concluded respectively as follows:

Firstly, a real-time data driven event detection method was presented, based on random matrix theory and Kalman filtering. In addition, as the data conditioner of the presented method, a dynamic Kalman filtering technique was developed through the adjustment of the measurement noise covariance matrix in order to effectively reduce noise and recover missing samples in PMU data. Both simulated and real PMU data are used to validate the presented method. The experimental results have shown that the dynamic Kalman filter outperforms the original one significantly, regardless of noise reduction or missing data recovery. Furthermore, the comparison results have demonstrated that the improved event detection method is much more robust than the original one, especially in practical situations where PMU data is collected with significantly heavy noise or plenty of missing samples.

Secondly, a short-term individual residential load forecasting method was proposed, based on deep learning and k-means clustering. Besides, a comprehensive feature expression strategy is utilised to provide the proposed method with sufficient information at each time step in order to improve the forecasting accuracy. A real-world residential load dataset from Ireland is used to verify the proposed method. The experimental results have shown that the proposed short-term residential load forecasting method has improved the forecasting accuracy significantly in terms of both RMSE and MAE, compared with the PDRNN. Moreover, the effect of k-means clustering on the proposed method is explored, and the comparison results have suggested that both the cluster distribution and overall loss of k-means clustering should be carefully considered for the purpose of improving the forecasting accuracy, when combining the clustering result with the LSTM models.

Thirdly, an online individual residential load forecasting method was developed, based on deep learning and dynamic mirror descent. A few modifications were made to the original DMD in order to enable it to become applicable to real-time residential load

forecasting. In addition, a comprehensive feature expression strategy is exploited to provide the developed method with sufficient information at each time step for the purpose of increasing the forecasting accuracy. A real-world Irish residential load dataset is employed to testify the developed method. The experimental results have demonstrated that the developed online residential load forecasting method has substantially improved the forecasting accuracy, regardless of RMSE or MAE, compared to the offline method. Besides, the effect of the parameter η of the modified DMD on the developed method is investigated, and the comparison results have indicated that an optimal value of the parameter η of the modified DMD is very likely to be found out to obtain the maximum forecasting accuracy improvement, when applying the modified DMD to real-time residential load forecasting.

6.2 Future Work

Although the thesis has contributed to big data analysis for smart grids primarily via deep learning, there are still a number of research gaps to bridge in the research field of applications of deep learning to smart grids. For example, it is commonly accepted that deep learning needs a huge amount of data to build a well-trained model so that it has desirable performance. However, in practice, only a small amount of data is available for training a deep learning model in most cases, which tends to lead to overfitting and have a significantly negative influence on the generalization ability of deep learning models. So, as a common solution to this problem, data augmentation is supposed to be adopted to enrich data. During the past few years, a variety of different data augmentation techniques have been developed and applied to diversify data, such as, rotating, flipping, blurring, scaling, translating, and cropping [159]. But, these techniques tend to be applicable to image processing. In other words, they can hardly be applied to process power system data, because these techniques were originally designed to process images but power system data cannot be organised as image data intrinsically. As a result, a more general data augmentation technique should be urgently developed to solve this issue.

Most recently, a novel smart data augmentation technique was presented to diversify data and improve the generalization capability of deep learning models [159]. To be specific, it aims to create a separate network to learn to generate augmented data during

the training process of a target network in a smart way in order to minimize the error of the target network and ensure its generalization capability. The separate network, namely the data augmenter, learns to merge two or more samples to generate a synthetic sample to feed to the target network for training. This smart data augmentation technique was originally developed for image processing, but its intrinsic idea is not confined to image data. In addition, it can be applied together with some traditional regularization methods to further improve the generalization ability of deep learning models, such as dropout and batch normalization. So, considering this fact, the research on applications of deep learning in smart grids will continue to be conducted in the near future from the following aspects:

Firstly, short-term residential load forecasting with a small dataset will be explored. As accurate residential load forecasting is able to effectively support demand response programs, its performance plays an essential role in demand side response. But, at present, many publicly available practical datasets of residential load do not have enough data for training, because they only include a short period of sampling. Hence, smart data augmentation is a potential method to enrich these datasets and obtain a satisfactory forecasting accuracy of deep leaning models.

Secondly, residential load classification with only a small amount of available data will be investigated. As a good understanding of residential energy consumption patterns is crucial for the provision of customized services to residents, precise residential load classification has a rather important influence on the quality of customized services. However, in practice, it is quite difficult to acquire a large number of load profiles from residents over a long time. Thus, smart data augmentation is a promising method to diversify residential load profiles and achieve a desirable classification accuracy of deep leaning models.

In summary, as this novel smart data augmentation technique is very likely to be effectively combined with deep learning, the future work will be focused on the above two research topics to contribute to the development of smart grids.

References

[1] C. P. Vineetha, and C. A. Babu, "Smart grid challenges, issues and solutions," in *2014 International Conference on Intelligent Green Building and Smart Grid (IGBSG)*, Taipei, Taiwan, Apr. 2014.

[2] Y. Guo, M. Pan, and Y. Fang, "Optimal power management of residential customers in the smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1593-1606, Sept. 2012.

[3] Q. Sun, J. Wu, N. Jenkins, and J. Ekanayake, "Comparison of the development of smart grids in China and the United Kingdom," in *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT)*, Gothenberg, Sweden, Oct. 2010.

[4] F. Xu, and L. Lai, "Scope design, characteristics and functionalities of Smart Grid," in *2011 IEEE Power and Energy Society General Meeting*, San Diego, CA, USA, July 2011.

[5] Y. Huang, S. Werner, J. Huang, N. Kashyap, and V. Gupta, "State estimation in electric power grids: Meeting new challenges presented by the requirements of the future grid," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 33-43, Sept. 2012.

[6] S. Khajeh, A. Attari, M. R. Shakarami, E. S. Pour, "Pareto optimal reconfiguration of power distribution systems using a genetic algorithm based on NSGA-II," in *2016 21st Conference on Electrical Power Distribution Networks Conference (EPDC)*, Karaj, Iran, Apr. 2016.

[7] P. Yang, G. Tang, and A. Nehorai, "A game-theoretic approach for optimal time-of-use electricity pricing," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 884-892, May 2013.

[8] E. Celebi, and J. D. Fuller, "Time-of-use pricing in electricity markets under different market structures," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1170-1181, Aug. 2012.

[9] P. Palensky, and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381-388, Aug. 2011.

[10] F. B. Beidou, W. G. Morsi, C. P. Diduch, and L. Chang, "Smart grid: Challenges, research directions and possible solutions," in *The 2nd International Symposium on Power Electronics for Distributed Generation Systems*, Hefei, China, June 2010.

[11] S. A. Arefifar, Y. A. I. Mohamed, and T. H. M. EL-Fouly, "Comprehensive operational planning framework for self-healing control actions in smart distribution grids," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4192-4200, Nov. 2013.

[12] D. Jia, X. Meng, and X. Song, "Study on technology system of self-healing control in smart distribution gird," in *2011 International Conference on Advanced Power System Automation and Protection*, Beijing, China, Oct. 2011.

[13] W. Tushar, B. Chai, C. Yuen, S. Huang, D. B. Smith, H. V. Poor, and Z. Yang, "Energy storage sharing in Smart Grid: A modified auction-based approach," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1462-1475, May 2016.

[14] O. Elma, and U. S. Selamogullari, "An overview of demand response applications under smart grid concept," in *2017 4th International Conference on Electrical and Electronic Engineering (ICEEE)*, Ankara, Turkey, Apr. 2017.

[15] J. Zhu, "Communication network for smart grid interoperability," in *2015 IEEE International Conference on Communication Software and Networks (ICCSN)*, Chengdu, China, June 2015.
[16] A. Farraj, E. Hammad, and D. Kundur, "A cyber-physical control framework for transient stability in smart grids," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1205-1215, Mar. 2018.
[17] H. Hu, Y. Wen, T. S. Chua, and X. Li, "Toward scalable systems for big data analytics: A technology tutorial," *IEEE Access*, vol. 2, pp. 652-687, June 2014.
[18] S. Athmaja, M. Hanumanthappa, and V. Kavitha, "A survey of machine learning algorithms for big data analytics," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, Mar. 2017.
[19] P. Goel, A. Datta, and M. S. Mannan, "Application of big data analytics in process safety and risk management," in *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, Dec. 2017.
[20] A. Sanchez, and Wilson Rivera, "Big data analysis and visualization for the smart grid," in *2017 IEEE International Congress on Big Data (BigData Congress)*, Honolulu, HI, USA, June 2017.
[21] S. M. A. Bhuiyan, J. F. Khan, and G. V. Murphy, "Big data analysis of the electric power PMU data from Smart Grid," in *SoutheastCon 2017*, Charlotte, NC, USA, Mar. 2017.
[22] L. Hou, Y. Zhang, Y. Yu, Y. Shi, and K. Liang, "Overview of data mining and visual analytics towards big data in smart grid," in *2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI)*, Beijing, China, Oct. 2016.
[23] C. Kang, Y. Wang, Y. Xue, G. Mu, and R. Liao, "Big data analytics in China's electric power industry: Modern information, communication technologies, and millions of smart meters," *IEEE Power and Energy Magazine*, vol. 16, no. 3, pp. 54-65, May 2018.
[24] J. Wu, K. Ota, M. Dong, J. Li, and H. Wang, "Big data analysis-based security situational awareness for Smart Grid," *IEEE Transactions on Big Data*, vol. 4, no. 3, pp. 408-417, Sept. 2018
[25] B. Li, M. C. Kisacikoglu, C. Liu, N. Singh, and M. E. Kantarci, "Big data analytics for electric vehicle integration in green smart cities," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 19-25, Nov. 2017.
[26] F. Luo, Z. Dong, J. Zhao, X. Zhang, W. Kong, and Y. Chen, "Enabling the big data analysis in the smart grid," in *2015 Power and Energy Society General Meeting*, Denver, CO, USA, July 2015.
[27] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Transactions on Smart Grid*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8322199.
[28] R. Moghaddass, and J. Wang, "A hierarchical framework for smart grid anomaly detection using large-scale smart meter data," *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 5820-5830, Nov. 2018.
[29] W. Wang, L. He, P. Markham, H. Qi, Y. Liu, Q. C. Cao, and L. M. Tolbert, "Multiple event detection and recognition through sparse unmixing for high-resolution situational awareness in power grid," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1654-1664, July 2014.
[30] P. Harrington, "Machine learning in action," Manning Publications, Apr. 2012.

[31] A. Saez, J. S. Monedero, P. A. Gutierrez, and C. H. Martinez, "Machine learning methods for binary and multiclass classification of melanoma thickness from dermoscopic images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 4, pp. 1036-1045, Apr. 2016.

[32] K. I. Kim, K. Jung, S. H. Park, and H. J. Kim, "Support vector machines for texture classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1542-1550, Nov. 2002.

[33] M. C. Ganiz, C. George, and W. M. Pottenger, "Higher order naive Bayes: A novel non-IID approach to text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1542-1550, July 2011.

[34] F. Ye, Z. Zhang, K. Chakrabarty, and X. Gu, "Board-level functional fault diagnosis using multikernel support vector machines and incremental learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 2, pp. 279-290, Feb. 2014.

[35] N. Buduma, "Fundamentals of deep learning," O'Reilly Media, June 2017.

[36] P. Zhou, H. Jiang, L. Dai, Y. Hu, and Q. Liu, "State-clustering based multiple deep neural networks modeling approach for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 4, pp. 631-642, Apr. 2015.

[37] M. Wang, Z. Ning, C. Xiao, and T. Li, "Sentiment classification based on information geometry and deep belief networks," *IEEE Access*, vol. 6, pp. 35206-35213, June 2018.

[38] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. D. Vries, M. J. N. L. Benders, and I. Isgum, "Automatic segmentation of MR brain images with a convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1252-1261, May 2016.

[39] A. Romero, C. Gatta, and G. C. Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349-1362, Mar. 2016.

[40] M. A. Uddin, J. B. Joolee, A. Alam, and Y. K. Lee, "Human action recognition using adaptive local motion descriptor in Spark," *IEEE Access*, vol. 5, pp. 21157-21167, Oct. 2017.

[41] Y. Peng, K. Chen, G. Wang, W. Bai, Y. Zhao, H. Wang, Y. Geng, Z. Ma, and L. Gu, "Towards comprehensive traffic forecasting in cloud computing: Design and application," *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 2210-2222, Aug. 2016.

[42] X. Liu, D. Zhao, L. Xu, W. Zhang, J. Yin, and X. Chen, "A distributed video management cloud platform using Hadoop," *IEEE Access*, vol. 3, pp. 2637-2643, Dec. 2015.

[43] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CESS Journal of Power and Energy Systems*, vol. 4, no. 3, pp. 362-370, Sept. 2018.

[44] X. Dong, L. Qian, and L. Huang, "Short-term load forecasting in smart grid: a combined CNN and k-means clustering approach," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju, South Korea, Feb. 2017.

[45] Y. He, J. Deng, and H. Li, "Short-term power load forecasting with deep belief network and copula models," in *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, Aug. 2017.

[46] S. Hosein, and P. Hosein, "Load forecasting using deep neural networks," in *2017 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, Apr. 2017.
[47] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, "Deep belief network based electricity load forecasting: An analysis of Macedonian case," *Energy*, vol. 115, part 3, pp. 1688-1700, Nov. 2016.
[48] J. F. Toubeau, J. Bottieau, F. Vallee, and Z. D. Greve, "Deep learning-based multivariate probabilistic forecasting for short-term scheduling in power markets," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1203-1215, Mar. 2019.
[49] T. Hossen, S. J. Plathottam, R. K. Angamuthu, P. Ranganathan, and H. Salehfar, "Short-term load forecasting using deep neural networks (DNN) ," in *2017 North American Power Symposium (NAPS)*, Morgantown, WV, USA, Sept. 2017.
[50] D. C. Mocanu, E. Mocanu, P. H. Nguyen, M. Gibescu, and A. Liotta, "Big IoT data mining for real-time energy disaggregation in buildings," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Budapest, Hungary, Oct. 2016.
[51] J. Kelly, and W. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," [Online]. Available: https://arxiv.org/pdf/1507.06594.pdf.
[52] H. Rafiq, H. Zhang, H. Li, and M. K. Ochani, "Regularized LSTM based deep learning model: First step towards real-time non-intrusive load monitoring," in *2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, Canada, Aug. 2018.
[53] K. Tornai, A. Olah, R. Drenyovszki, L. Kovacs, I. Pinte, and J. Levendovszky, "Recurrent neural network based user classification for smart grids," in *2017 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, Apr. 2017.
[54] Y. Wang, Q. Chen, D. Gan, J. Yang, D. S. Kirschen, and C. Kang, "Deep learning-based socio-demographic information identification from smart meter data," *IEEE Transactions on Smart Grid*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8291011.
[55] W. Kong, Z. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 1087-1088, Jan. 2018.
[56] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. G. Slootweg, "On-line building energy optimization using deep reinforcement learning," *IEEE Transactions on Smart Grid*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8356086.
[57] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3259-3269, July 2018.
[58] X. Qu, X. Kang, C. Zhang, S. Jiang, and X. Ma, "Short-term prediction of wind power based on deep long short-term memory," in *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Xi'an, China, Oct. 2016.
[59] J. Yan, H. Zhang, Y. Liu, S. Han, L. Li, and Z. Lu, "Forecasting the high penetration of wind power on multiple scales using multi-to-multi mapping," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3276-3284, May 2018.
[60] A. Gensler, J. Henze, B. Sick, and N. Raabe, "Deep learning for solar power forecasting – An approach using autoencoder and LSTM neural networks," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*,

Budapest, Hungary, Oct. 2016.
[61] J. I. Lee, I. W. Lee, and S. H. Kim, "Multi-site photovoltaic power generation forecasts based on deep-learning algorithm," in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, South Korea, Oct. 2017.
[62] Y. Q. Neo, T. T. Teo, W. L. Woo, T. Logenthiran, and A. Sharma, "Forecasting of photovoltaic power using deep belief network," in *TENCON 2017 - 2017 IEEE Region 10 Conference*, Penang, Malaysia, Nov. 2017.
[63] Z. Zhao, G. Xu, Y. Qi, N. Liu, and T. Zhang, "Multi-patch deep features for power line insulator status classification from aerial images," in *2016 international Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, July 2016.
[64] Y. Ma, Z. Guo, J. Su, Y. Chen, X. Du, Y. Yang, C. Li, Y. Lin, and Y. Geng, "Deep learning for fault diagnosis based on multi-sourced heterogeneous data," in *2014 International Conference on Power System Technology*, Chengdu, China, Oct. 2014.
[65] S. Wang, S. Fan, J. Chen, X. Liu, B. Hao, and J. Yu, "Deep-learning based fault diagnosis using computer-visualised power flow," *IET Generation, Transmission, and Distribution*, vol.12, no. 17, pp. 3985-3992, Sept. 2018.
[66] Y. Wang, M. Liu, and Z. Bao, "Deep learning neural network for power system fault diagnosis," in *2016 35th Chinese Control Conference (CCC)*, Chengdu, China, July 2016.
[67] B. Bhattacharya, and A. Sinha, "Intelligent fault analysis in electrical power grids," in *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, Boston, MA, USA, Nov. 2017.
[68] E. Balouji, and O. Salor, "Classification of power quality events using deep learning on event images," in *2017 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, Shahrekord, Iran, Apr. 2017.
[69] N. Mohan, K. P. Soman, and R. Vinayakumar, "Deep power: Deep learning architectures for power quality disturbances classification," in *2017 International Conference on Technological Advancements in Power and Energy (TAP Energy)*, Kollam, India, Dec. 2017.
[70] Z. Yan, and Y. Xu, "Data-driven load frequency control for stochastic power systems: A deep reinforcement learning method with continuous action search," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1653-1656, Mar. 2019.
[71] M. Sun, I. Konstantelos, and G. Strbac, "A deep learning-based feature extraction framework for system security assessment," *IEEE Transactions on Smart Grid*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8477148.
[72] E. D. Varga, S. F. Beretka, C. Noce, and G. Sapienza, "Robust real-time load profile encoding and classification framework for efficient power systems operation," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1897-1904, July 2015.
[73] J. Zhang, C. Lu, C. Fang, X. Ling, and Y. Zhang, "Load shedding scheme with deep reinforcement learning to improve short-term voltage stability," in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, Singapore, May 2018.
[74] B. Tan, J. Yang, X. Pan, J. Li, P. Xie, and C. Zeng, "Representational learning approach for power system transient stability assessment based on convolutional neural network," *The Journal of Engineering*, vol. 2017, no. 13, pp. 1847-1850,

2017.
[75] L. Zheng, W. Hu, Y. Zhou, Y. Min, X. Xu, C. Wang, and R. Yu, "Deep belief network based nonlinear representation learning for transient stability assessment," in *2017 Power and Energy Society General Meeting*, Chicago, IL, USA, July 2017.
[76] J. Hou, C. Xie, T. Wang, Z. Yu, Y. Lu, and H. Dai, "Power system transient stability assessment based on voltage phasor and convolutional neural network," in *2018 IEEE International Conference on Energy Internet (ICEI)*, Beijing, China, May 2018.
[77] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in Smart Grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2505-2516, Sept. 2017.
[78] J. Wei, and G. J. Mendis, "A deep learning-based cyber-physical strategy to mitigate false data injection attack in smart grids," in *2016 Joint Workshop on Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG)*, Vienna, Austria, Apr. 2016.
[79] J. Wang, D. Shi, Y. Li, J. Chen, H. Ding, and X. Duan, "Distributed framework for detecting PMU data manipulation attacks with deep autoencoders," *IEEE Transactions on Smart Grid*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8418851.
[80] H. Wang, J. Ruan, G. Wang, B. Zhou, Y. Liu, X. Fu, and J. Peng, "Deep learning-based interval state estimation of AC smart grids against sparse cyber attacks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4766-4778, Nov. 2018.
[81] Z. Zheng, Y. Yang, X. Niu, H. Dai, and Y. Zhou, "Wide and deep convolutional neural networks for electricity-theft detection to secure smart grids," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1606-1615, Apr. 2018.
[82] P. Zeng, H. Li, H. He, and S. Li, "Dynamic energy management of a microgrid using approximate dynamic programming and deep recurrent neural network learning," *IEEE Transactions on Smart Grid*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8419206.
[83] T. Chen, and W. Su, "Local energy trading behavior modeling with deep reinforcement learning," *IEEE Access*, vol. 6, pp. 62806-62814, Oct. 2018.
[84] S. S. Mousavi-Seyedi, F. Aminifar, and S. Afsharnia, "Application of WAMS and SCADA data to online modeling of series-compensated transmission lines," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1968-1976, Jul. 2017.
[85] V. Kekatos, G. B. Giannakis, and B. Wollenberg, "Optimal placement of phasor measurement units via convex relaxation," *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1521-1530, Aug. 2012.
[86] J. Liu, J. Tang, F. Ponci, A. Monti, C. Muscas, and P. A. Pegoraro, "Trade-offs in PMU deployment for state estimation in active distribution grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 915-924, Jun. 2012.
[87] Q. Li, T. Cui, Y. Weng, R. Negi, F. Franchetti, and M. D. Ilic, "An information-theoretic approach to PMU placement in electric power systems," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 446-456, Mar. 2013.
[88] T. K. Maji and P. Acharjee, "Multiple solutions of optimal PMU placement using exponential binary PSO algorithm for smart grid applications," *IEEE Transactions on Industry Applications*, vol. 53, no. 3, pp. 2550-2559, May 2017.
[89] W. Lin, C. Yang, J. Lin, and M. Tsay, "A fault classification method by RBF neural network with OLS learning procedure," *IEEE Transactions on Power*

Delivery, vol. 16, no. 4, pp. 473-477, Oct. 2001.
[90] S. Vasilic and M. Kezunovic, "Fuzzy ART neural network algorithm for classifying the power system faults," *IEEE Transactions on Power Delivery*, vol. 20, no. 2, pp. 1306-1314, Apr. 2005.
[91] D. Nguyen, R. Barella, S. A. Wallace, X. Zhao, and X. Liang, "Smart grid line event classification using supervised learning over PMU data streams," in *2015 Sixth International Green and Sustainable Computing Conference (IGSC)*, Las Vegas, NV, USA, Dec. 2015.
[92] U. Adhikari, T. Morris, and S. Pan, "Applying Hoeffding adaptive trees for real-time cyber-power event and intrusion classification," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4049-4060, Sept. 2018.
[93] L. Xie, Y. Chen, and P. R. Kumar, "Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis," *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 2784-2794, Nov. 2014.
[94] M. Rafferty, X. Liu, D. M. Laverty, and S. McLoone, "Real-time multiple event detection and classification using moving window PCA," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2537-2548, Sept. 2016.
[95] Y. Guo, K. Li, D. M. Laverty, and Y. Xue, "Synchrophasor-based islanding detection for distributed generation systems using systematic principal component analysis approaches," *IEEE Transactions on Power Delivery*, vol. 30, no. 6, pp. 2544-2552, Dec. 2015.
[96] Y. Ge, A. J. Flueck, D. Kim, J. Ahn, J. Lee, and D. Kwon, "Power system real-time event detection and associated data archival reduction based on synchrophasors," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 2088-2097, July 2015.
[97] Y. Guo, K. Li, D. M. Laverty, and Y. Xue, "Synchrophasor-based islanding detection for distributed generation systems using systematic principal component analysis approaches," *IEEE Transactions on Power Delivery*, vol. 30, no. 6, pp. 2544-2552, Dec. 2015.
[98] D. Kim, T. Chun, S. Yoon, G. Lee, and Y. Shin, "Wavelet-based event detection method using PMU data," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1154-1162, May 2017.
[99] R. G. Kavasseri, Y. Cui, and S. M. Brahma, "A new approach for event detection based on energy functions," in *Proc. IEEE PES General Meeting*, 2014, pp. 1-5.
[100] Z. Lin, T. Xia, Y. Ye, Y. Zhang, L. Chen, Y. Liu, K. Tomsovic, T. Bilke, and F. Wen, "Application of wide area measurement systems to islanding detection of bulk power systems," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 2006-2015, May 2013.
[101] X. Liang, S. A. Wallace, and D. Nguyen, "Rule-based data-driven analytics for wide-area fault detection using synchrophasor data," *IEEE Transactions on Industry Applications*, vol. 53, no. 3, pp. 1789-1798, May 2017.
[102] X. He, R. C. Qiu, Q. Ai, Y. Cao, J. Gu, and Z. Jin, "A random matrix theoretical approach to early event detection in smart grids," Sept. 2015. [Online]. Available: https://arxiv.org/pdf/1502.00060.pdf.
[103] J. Zhang, G. Welch, G. Bishop, and Z. Huang, "A two-stage Kalman filter approach for robust and real-time power system state estimation," *IEEE Transactions on Sustainable Energy*, vol. 5, no. 2, pp. 629-636, Apr. 2014.
[104] P. Gao, M. Wang, S. G. Ghiocel, J. H. Chow, B. Fardanesh, and G. Stefopoulos, "Missing data recovery by exploiting low-dimensionality in power system synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 31, no.

2, pp. 1006-1013, Mar. 2016.
[105] X. Xu, X. He, Q. Ai, and R. C. Qiu, "A correlation analysis method for power systems based on random matrix theory," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1811-1820, Jul. 2017.
[106] X. He, Q. Ai, R. C. Qiu, W. Huang, L. Piao, and H. Liu "A big data architecture design for smart grids based on random matrix theory," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 674-686, Mar. 2017.
[107] H. Karimipour, and V. Dinavahi, "Extended Kalman filter-based parallel dynamic state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1539-1549, May 2015.
[108] M. Netto, J. Zhao, and L. Mili, "A robust extended Kalman filter for power system dynamic state estimation using PMU measurements," in *2016 IEEE Power and Energy Society General Meeting*, Boston, USA, Jul. 2016.
[109] E. Ghahremani, and I. Kamwa, "Local and wide-area PMU-based decentralized dynamic state estimation in multi-machine power systems," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 547-562, Jan. 2016.
[110] K. D. Jones, A. Pal, and J. S. Thorp, "Methodology for performing synchrophasor data conditioning and validation," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1121-1130, May 2015.
[111] F. Gao, J. S. Thorp, A. Pal, and S. Gao, "Dynamic state prediction based on auto-regression model using PMU data," in *Proc. IEEE Power and Energy Conference at Illinois (PECI)*, 2012, pp. 1-5.
[112] F. Mahmood, H. Hooshyar, and L. Vanfretti, "Extracting steady state components from synchrophasor data using Kalman filters," *energies*, vol. 9, no. 5, pp. 315-329, Apr. 2016.
[113] 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, Feb. 2011.
[114] P. M. Ashton, C. S. Saunders, G. A. Taylor, A. M. Carter, and M. E. Bradley, "Inertia estimation of the GB power system using synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 30, no. 2, pp. 701-709, March 2015.
[115] P. M. Ashton, G. A. Taylor, M. R. Irving, I. Pisica, A. M. Carter, and M. E. Bradley, "Novel application of detrended fluctuation analysis for state estimation using synchrophasor measurements," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1930-1938, May 2013.
[116] B. Huang, D. Wu, C. S. Lai, X. Cun, H. Yuan, F. Xu, L. L. Lai, and K. F. Tsang, "Load forecasting based on deep long short-term memory with consideration of costing correlated factor," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, Porto, Portugal, July 2018.
[117] Y. Wang, Q. Chen, N. Zhang, and Y. Wang, "Conditional residual modeling for probabilistic load forecasting," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7327-7330, Nov. 2018.
[118] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 4, pp. 438-447, July 2010.
[119] Y. Wang, Q. Xia, and C. Kang, "Secondary forecasting based on deviation analysis for short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp. 500-507, May 2011.

[120] S. Fan, and L. Chen, "Short-term load forecasting based on an adaptive hybrid method," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 392-401, Feb. 2006.
[121] S. Fan, K. Methaprayoon, and W. J. Lee, "Multiregion load forecasting for system with large geographical area," *IEEE Transactions on Industry Applications*, vol. 45, no. 4, pp. 1452-1459, July 2009.
[122] V. H. Hinojosa, and A. Hoese, "Short-term load forecasting using fuzzy inductive reasoning and evolutionary algorithms," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 565-574, Feb. 2010.
[123] Y. Chen, P. B. Luh, C. Guan, Y. Zhao, L. D. Michel, M. A. Coolbeth, P. B. Friedland, and S. J. Rourke, "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 322-330, Feb. 2010.
[124] N. Amjady, "Short-term bus load forecasting of power systems by a new hybrid method," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 333-341, Feb. 2007.
[125] D. Liu, L. Zeng, C. Li, K. Ma, Y. Chen, and Y. Cao, "A distributed short-term load forecasting method based on local weather information," *IEEE Systems Journal*, vol. 12, no. 1, pp. 208-215, Mar. 2018.
[126] Y. Zhang, Q. Zhou, C. Sun, S. Lei, Y. Liu, and Y. Song, "RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 853-858, Aug. 2008.
[127] X. Sun, P. B. Luh, K. W. Cheung, W. Guan, L. D. Michel, S. S. Venkata, and M. T. Miller, "An efficient approach to short-term load forecasting at the distribution level," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2526-2537, July 2016.
[128] S. A. Villalba, and C. A. Bel, "Hybrid demand model for load estimation and short term load forecasting in distribution electric systems," *IEEE Transactions on Power Delivery*, vol. 15, no. 2, pp. 764-769, Apr. 2000.
[129] J. W. Taylor, and R. Buizza, "Neural network load forecasting with weather ensemble predictions," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 626-632, Aug. 2002.
[130] K. N. Filho, A. D. P. Lotufo, and C. R. Minussi, "Short-term multinodal load forecasting using a modified general regression neural network," *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2862-2869, Oct. 2011.
[131] B. Zhang, X. Xu, H. Xing, and Y. Li, "A deep learning based framework for power demand forecasting with deep belief networks," in *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, Taipei, Taiwan, Dec. 2017.
[132] S. Hosein, and P. Hosein, "Load forecasting using deep neural networks," in *2017 IEEE Power and Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, Apr. 2017.
[133] J. Bedi, and D. Toshniwai, "Empirical mode decomposition based deep learning for electricity demand forecasting," *IEEE Access*, vol. 6, pp. 49144-49156, Aug. 2018.
[134] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8372953.
[135] L. Han, Y. Peng, Y. Li, B. Yong, Q. Zhou, and L. Shu, "Enhanced deep networks

for short-term and medium-term load forecasting," *IEEE Access*, vol. 7, pp. 4045-4055, Dec. 2018.
[136] W. Kong, Z. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y.Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841-851, Jan. 2019.
[137] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting – A novel pooling deep RNN," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271-5280, Sept. 2018.
[138] F. P. Anaraki, and S. Becker, "Preconditioned data sparsification for big data with applications to PCA and k-means," *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2954-2974, May 2017.
[139] F. L. Quilumba, W. J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 911-918, Mar. 2015.
[140] Electricity smart metering customer behaviour trials (CBT) findings report. [Online]. Available: https://www.cru.ie/wp-content/uploads/2011/07/cer11080ai.pdf.
[141] G. Xie, X. Chen, and Y. Weng, "An integrated Gaussian process modeling framework for residential load prediction," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7238-7248, Nov. 2018.
[142] I. A. Sajjad, G. Chicco, and R. Napoli, "Definitions of demand flexibility for aggregate residential loads," *IEEE Transactions on Smart Grid*, vol. 7, no. 6, pp. 2633-2643, Nov. 2016.
[143] D. Zhou, M. Balandat, and C. Tomlin "A Bayesian perspective on residential demand response using smart meter data," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, USA, Sept. 2016.
[144] T. K. Wijaya, M. Vasirani, S. Humeau, and K. Aberer, "Cluster-based aggregate forecasting for residential electricity demand using smart meter data," in *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, USA, Nov. 2015.
[145] S. Humeau, T. K. Wijaya, M. Vasirani, and K. Aberer, "Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households," in *2013 Sustainable Internet and ICT for Sustainability (SustainIT)*, Palermo, Italy, Oct. 2013.
[146] A. Marinescu, I. Dusparic, Colin Harris, Vinny Cahill, and S. Clarke, "A dynamic forecasting method for small scale residential electrical demand," in *2014 International Joint Conference on Neural Networks (IJCNN)*, Beijing, China, July 2014.
[147] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1591-1598, July 2017.
[148] A. Marinescu, C. Harris, I. Dusparic, S. Clarke, and V. Cahill, "Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods," in *2013 2nd International Workshop on Software Engineering Challenges for the Smart Grid (SE4SG)*, San Francisco, CA, USA, May 2013.
[149] B. P. Campos, and M. R. D. Silva, "Demand forecasting in residential distribution feeders in the context of smart grids," in *2016 12th IEEE International*

Conference on Industry Applications (INDUSCON), Curitiba, Brazil, Nov. 2016.
[150] J. Zheng, X. Chen, K. Yu, L. Gan, Y. Wang, and K. Wang, "Short-term power load forecasting of residential community based on GRU neural network," in *2018 International Conference on Power System Technology (POWERCON)*, Guangzhou, China, Nov. 2018.
[151] J. Xu, M. Yue, D. Katramatos, and S. Yoo, "Spatial-temporal load forecasting using AMI data," in *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Sydney, NSW, Australia, Nov. 2016.
[152] C. Dinesh, S. Makonin, and I. V. Bajie, "Residential power forecasting using load identification and graph spectral clustering," *IEEE Transactions on Circuits and Systems II*: *Express Briefs*. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8606124.
[153] J. Vossen, B. Feron, and A. Monti, "Probabilistic forecasting of household electrical load using artificial neural networks," in *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, Boise, ID, USA, June 2018.
[154] D. Gan, Y. Wang, N. Zhang, and W. Zhu, "Enhancing short-term probabilistic residential load forecasting with quantile long-short-term memory," *The Journal of Engineering*, vol. 2017, no. 14, pp. 2622-2627, 2017.
[155] T. Hossen, A. S. Nair, R. A. Chinnathambi, and P. Ranganathan, "Residential load forecasting using deep neural networks (DNN)," in *2018 North American Power Symposium (NAPS)*, Fargo, ND, USA, Sept. 2018.
[156] W. Kong, Z. Dong, F. Luo, K. Meng, W. Zhang, F. Wang, and X. Zhao, "Effect of automatic hyperparameter tuning for residential load forecasting via deep learning," in *2017 Australasian Universities Power Engineering Conference (AUPEC)*, Melbourne, VIC, Australia, Nov. 2017.
[157] E. C. Hall, and R. M. Willett, "Online convex optimization in dynamic environments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 647-662, June 2015.
[158] G.S. Ledva, L. Balzano, and J. L. Mathieu, "Real-time energy disaggregation of a distribution feeder's demand using online learning," *IEEE Transactions on Power Systems*, vol. 33, no. 5, pp. 4730-4740, Sept. 2018.
[159] J. Lemley, S. Barzrafkan, and P. Corcoran, "Smart augmentation learning an optimal data augmentation strategy," *IEEE Access*, vol. 5, pp. 5858-5869, Apr. 2017.