DISSERTATION

POWER SYSTEM DATA CLASSIFICATION AND PREDICTION BY FUNCTIONAL DATA ANALYSIS

Submitted by

Hongfei Sun

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2021

Doctoral Committee:

Advisor: Liuqing Yang
Co-Advisor: Jie Luo

Hongming Zhang
Dongliang Duan
Haonan Wang

ABSTRACT

POWER SYSTEM DATA CLASSIFICATION AND PREDICTION BY FUNCTIONAL DATA ANALYSIS

The last couple of decades have witnessed the development of our electric power grid. The growing population size and increasing consumerism have increased the load demand and brought more pressure on the grid. Meanwhile, new elements are being introduced to the power grid, such as various forms of renewable energy resources, electric vehicles, and so on, which need to be monitored constantly and managed adequately. In addition, the allocation of the various resources in the power systems is now conducted in a much more dynamic manner than ever. All these new dimensions have driven the development of the traditional grid into the smart grid and call for new methodologies in system design, operation, and control. This dissertation focuses on modeling power systems with data-driven approaches, with applications in power system cyber-attack detection and recovery, and large-scale, long-term load characterization. Firstly, the modeling of the spatial-temporal relationship among the quantities across the entire power systems is provided with applications to cyber-attack detection and data recovery. Then, the non-conforming load classification approaches based on Functional Principle Component Analysis (FPCA) will be introduced. This work is the first effort towards such loads due to the recently growing penetration of Distributed Energy Resources (DER) users. Lastly, we will introduce the regional high-resolution medium-term load forecasting approach. In order to satisfy the new purpose of load forecasting, serving for real-time applications, our approach can provide higher resolution than existing long-term load forecasting and longer leading time than the existing short-term load forecasting time-series load curve. Based on the presented case studies and simulation results, we provided the corresponding suggestions to the present industrial power system.

ACKNOWLEDGEMENTS

Coming from a different background, I started to pursuit my Ph.D. degree in power systems. Before that, I have been worked in an entirely different research field for one and a half years, but due to insufficient funding, I was almost forced to terminate my studies. By chance, I mustered up the courage to make the final attempt. During the entire process, I would like first to thank God Father, Lord Jesus Christ, and Holy Spirit, who gives me courage, wisdom, and strength, especially when I felled into helpless despair.

I want to sincerely express my gratitude to my advisor Dr. Liuqing Yang, who provided a chance to let me stay, has spent her efforts and time for the advisory. Her encouragements and supports are tremendous and critical. I would also like to appreciate Dr. Hongming Zhang, who guided me with his experience in industrial power grids, also provided care and encouragement for me. I would also like to appreciate my co-advisor Dr. Jie Luo, who provided abundance active supports and guidances. I would like to thank my outside committee member Dr. Haonan Wang, who has provided strong support on my researches. I would also like to appreciate my committee member Dr. Dongliang Duan, who has taught me lots of abilities that a Ph.D. needs.

I would also like to appreciate Prof. Maciejewski who is the first one to teach me how to do research. I would also like to appreciate the authors of minniWECC simulation system, Dr. Dan Trudnowski and Dr. John Undrill. I would also like to thank the manager of my internship of NREL, Dr. Bryan Palmintier.

I am also very grateful for my friends and colleagues: Cheng Guo, Xinhu Zheng, Shijian Gao, Dr. Dexin Wang, Dr. Luoyang Fang, Dr. Biyun Xie, Dr. Yajing Liu, Xinyi Lu, Wanxiao Xu, Mengting Lin, Dr. Sanjay Hosur, Yan Tang, Yufeng Jiao, Xuxing Zhang, and Yudi Cao.

Finally, I am grateful to my parents, Zhongwei Sun and Yanfeng Shang, for their continuous supports.

DEDICATION

To my Lord, God Father, Jesus Christ, Holy Spririt

and my parents, Zhongwei Sun and Yanfeng Shang

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Our current electric power grid was first built back in the 1890s and had been getting updates and improvements over time as technologies advance [2]. In the recent two decades, our power grid is being pushed to do more than it was initially designed to do, and more and more limitations of the aged infrastructure show up in terms of its size, scale, security, and adaptability [3]. The growing population and the increasing consumers bring more and more pressure and tremendous demand into the power system. Meanwhile, new elements are introduced to the power grid, such as the growing penetrations of renewable energy resources. These elements need to be monitored constantly and managed adequately. In addition, the allocation of the various resources in the power systems is now conducted more dynamically than in the past. All these new dimensions have driven the development of the traditional grid into the so-termed “smart grid” [2], which integrates the physical power grid with many cyber components, such as two-way communication, intelligent control systems, and high-performance computer processing, etc. [4].

On the one hand, the evolution of the traditional grid to the smart grid brought various opportunities with these new demands and infrastructures on the horizon. For example, it enables multiple new products, services, and markets and provides the power quality for various ranges of needs. On the other hand, it invites new challenges such as large-scale system modeling, centralized management for optimal operation, stable and secure wide-area sensing and communication, and criticality of data delivery, and so on [5].

In recent years, the data-driven approach has been popular in the power systems research fields. It is mainly due to operating the energy management system (EMS) and the distribution management system (DMS). Each utility control room has collected a large amount of data. The resolution of these data ranges from minutes’ intervals in the state estimation (SE) to 10-second intervals of the inter-control center communications protocol (ICCP) measurements. The rich data and information should be sufficient to support new technologies such as machine learning, big data approach, and functional data analysis. The new approach will enable utilities to accurately manage real-time operation and planning horizons to meet emerging challenges and needs driven by the "green energy" wave.

Among these new challenges encountered in the development of the modern power grid, this dissertation focuses on the classification and prediction of power systems based on functional data analysis with applications cyber-attack detection and recovery, and large-scale, long-term load characterization, shown as Figure 1.1. A model of power systems was first proposed based on functional data analysis. Then, an anomaly detection scheme (classification) was designed based on the proposed model. Based on the anomaly detection scheme, some types of cyber-attack were detected and recovered (prediction). For load characterization, we studied the non-conforming load identification (classification) problem and medium-term load forecasting (prediction) problem. Finally, based on the results, we provided some reasonable suggestions to the power systems industry in this dissertation.

1.1 Cyber-Security Problems

In the modern power grid, electronic components have replaced or are gradually replacing the original mechanical components. This introduces new modules that need real-time communications, such as Supervisory Control and Data Acquisition (SCADA) [6], Phasor Measurement Units (PMUs) and Wide-Area Measurements Systems (WAMS) [7]. In addition, the control, measurement, and communication processes are now conducted in an automated manner to ensure efficient and reliable power grid operations [3]. All these data collection modules using communication links are exposed to cyber security risks [8]. This is evident by the fact that the power system has become a prime target for cyber-criminals in recent years. Department of Energy (DoE) has reported 150 successful cyber attacks to our power grid between 2010 and 2014, and the number is surged by 380% between 2014 and 2015. The motivation for attacks includes geopolitics, sabotage, and financial reasons [9].

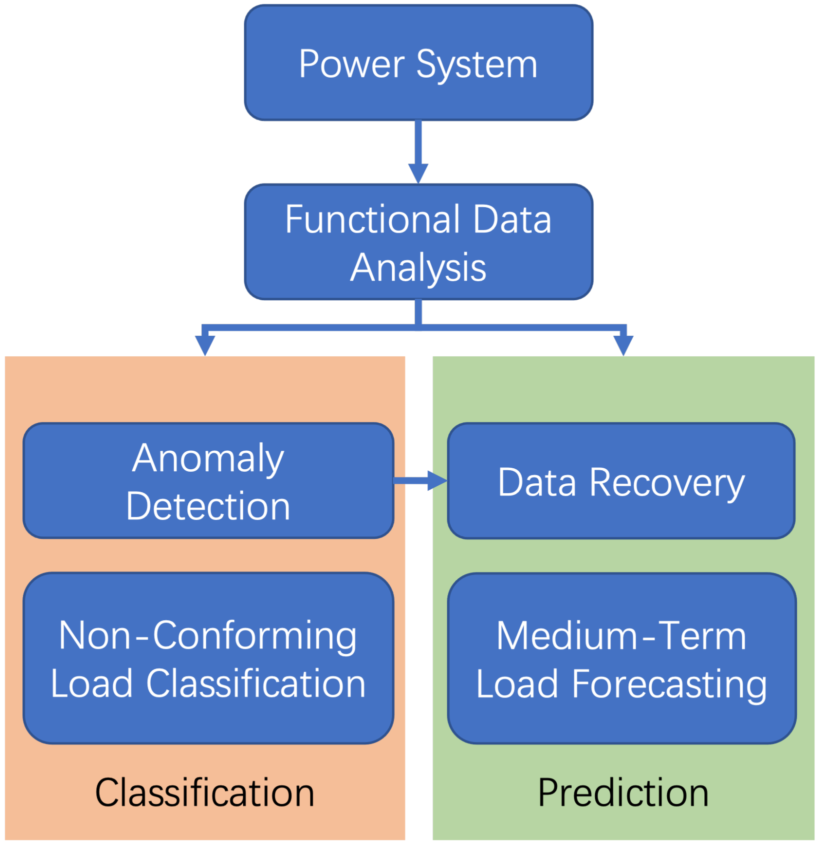

Figure 1.1: Research overview. This dissertation focuses on the classification and prediction of power systems based on functional data analysis with applications cyber-attack detection and recovery, and large-scale long-term load characterization.

Cyber-attacks are ubiquitous and unpredictable because they are subtly designed to be deceptive. In addition, it can be conducted from considerable remote locations. Therefore, the situational awareness modules for power systems which are heavily relied on cyber components need to be improved continuously even though they have been studied and continuously improved for decades [10, 11]. The cyber attacks to power systems could be organized into three different categories based on the objectives of the attacker [12]: 1) availability attack [13]; 2) integrity attack [14–16]; and 3) Confidentiality attack [17, 18]. The availability attacks are also called the denial of service (DoS) attacks and they will cause the unavailability of the critical power system measurement data since they try to prevent, interrupt or delay the communications in the power systems. The integrity attacks try to illegally disrupt the data exchange, for example, the false data injection (FDI) attacks. The FDI attacks refer to the case when an attacker compromises sensor readings in such a tricky way that undetected errors are introduced into the data during the calculations of state, variables, and values. One of FDI attacks that is the most detrimental to power systems is the stealthy attack, which will bypass the conventional bad data detection (BDD) to alter the results of the state estimation process. The confidentiality attacks attempt to obtain unauthorized information from network resources. In general, it is crucial to detect cyber attacks in modern power system and estimate the state of the system as accurate as possible to recover from the attacks. Considering that the confidentiality is quite different from the other two types of attacks in the sense that it would not affect the data collection process for power system monitoring, this research mainly focuses on the first two categories of cyber-attack, namely the availability and integrity attacks.

In recent years, there are various solutions proposed at different layers of the 7-layers OSI model of computer networking for cyber attacks in power systems. For example, at the network layer, [19] detects the occurrence of abnormal behaviors by modeling the temporal characteristics of the footprints left on the server by operation events; [20] models the normal data exchange flows and control operations to classify the normal and abnormal conditions. However, even though these methods can detect cyber attacks, the attacked data cannot be recovered. At the application layer, solutions are proposed for some specific applications, such as [8, 21, 22]. They focus on the State Estimation (SE) process and check the error between system estimates and system measurements to determine whether the system is under normal operation. However, these application-oriented approaches usually have high pertinence but low generality. Research at the data link layer is the current mainstream. They directly target at the data collection modules such as PMUs [7, 13, 23–26]. Besides high generality, these approaches directly model the measurement data, which makes it more possible for the grid to recover the normal data and state from attacks. In this dissertation, we also try to address the cyber security issues at the data link layer.

For this problem, we first propose a new data-driven system model to model the spatial-temporal relationship among quantities such as voltages and currents across the entire power systems. This spatial-temporal model can not only handle the dynamic feature of power systems but also handle the potential high non-linearity introduced by the renewable resources, power electronic devices, complex load components etc. which are now increasingly integrated into our grid. Secondly, we create an online anomaly detection scheme based on the proposed system model. It can detect in real time whether the measurement data on any bus of the power systems are normal at each moment by checking whether they satisfy the potential relationship captured by the proposed model both spatially and temporally. Then, we apply the anomaly detection scheme to the DoS attack, and try to detect and recover from the DoS attack with good accuracy. In addition, we consider a new FDI attack type which has never been considered in previous works but could be easily implemented by attackers and threaten the power system. It is the historical data injection (HDI), which uses the historical normal data as false data to be injected to the measurement data. Most existing works would fail to detect this type of attack since they do not take into account the temporal relationship among the data.

In order to show the generality of our model and the possibility to recover attacked data from our model, we first conduct simulations to check the fitness performance of the proposed system model for the ambient environment data. Results show that our model has a good performance for fitting the ambient data with small RMSE. Then, two case studies are conducted to show different applications of our approach: 1) the detection of and the data recovery from a DoS attack; and 2) the detection of the historical normal data injection attack. For the DoS attack case study, the proposed scheme has successfully detected and recovered from a DoS attack on a single bus, and it has a high accuracy of recovery with a very small error. As the number of attacked buses increases, the average recovery error increases as expected, but the normalized RMSE is still very small (at the order of 10^{-5}) even when 70 out of 121 buses were attacked. For the FDI attack case study, the proposed algorithm has successfully detected the historical FDI attack. It also shows that when an FDI attack occurs, the estimated value for other buses will also be impacted. As a result, the current scheme cannot recover measurement data from the FDI attack so far. However, at least the system would know the data are not authentic and would not take false operations based upon the wrong data. In the future, we would look into the data recovery by modifying our model to take buses with only one of them attacked at a time.

1.2 Non-Conforming Load Classification Problem

In the power system, the consumption patterns of individual customers (residential, commercial, industrial and agricultural, etc.) are different, and even the same types of customers may have different patterns [27]. Load classification is a significant part of load modeling in power systems, so the accuracy of load classification will directly affect the rationality of load modeling [28]. In addition, discovering different customers based on load classification can not only deepen people's understanding of the power system but also help utilities provide personalized power system planning and control decisions [29].

The concept of the non-conforming load was first proposed in [30], which was defined as the kind of individual load with the characteristics of irregular changes. At that time, the non-conforming load was mainly composed of industrial loads, and now the non-conforming loads are greatly expanded due to increased distributed energy resource (DER) users. The irregular changing characteristics of non-conforming loads usually bring high risks and various challenges concerning the stability of the power grid operation. For example, the industrial loads may change (increase or decrease) a considerable amount of load in the power system during a brief period. The DER customers may show a state of occasional failure at the observation end of the grid control center because of the intermittent switching to the power supply network or renewable energy (even if they may be consistent), which makes this part of the load in the power system discontinuous and causes frequent sudden changes. Therefore, the analysis and modeling research for the non-conforming load is needed. A suitable automatic classification algorithm is the first step of further research due to the massive amount of individual loads. However, so far, few researchers are working on this non-conforming load field because of two main reasons: 1) In the past, when the types and quantities of loads in the power system were relatively small, the demand for automatic classification of individual loads was not so urgent. Relying on engineers' direct observations to roughly classify individual loads is enough to meet the needs. 2) In the era of modern power grids, the growing number and diverse loads and the penetration of DER have made the demand for classification research of non-compliant loads more complex and severe. In recent years, various measurement and monitoring mechanisms in modern power grids, such as EMS and DMS, have brought a large amount and high-resolution measurement data that makes it possible to automatically and intelligently meet the load classification.

Given the importance of non-conforming loads in power systems and the lack of investigation, this research is the first effort towards the automatic classification/identification of such loads. Because the power grid can cover an extensive range, the entire area can be divided into different sub-regions based on geography or climate because climate always significantly affects people's lifestyles. Customers in the same area often have similar characteristics to some extent. This research makes the classification of non-conforming load based on quantifying the difference between the features of the individual load in a sub-region and the features of the aggregated load in the sub-region. One of the difficulties of this research is that the definition of the non-conforming load is vague, which means data are not labeled. Therefore, this research also summarizes three properties that need to be considered when analyzing non-conforming load problems in general based on the experience of experts of power systems. Then based on the reasonable usage of three properties and functional principal component analysis (FPCA) [31], a series of steps of dimension reduction, feature extraction, clustering analysis, and classification for non-conforming load or conforming load is gradually implemented.

Considering that the non-conforming extent of individual loads may change over time (e.g., DER users), this research proposes two classification processors: 1) the segmental dataset processor and 2) the full dataset processor. The former can analyze and classify the non-conforming characteristics of each individual load at some specific periods of the entire observation window. The latter can analyze and classify the non-conforming characteristics of the overall performance of each individual load during the whole observation window.

Finally, the proposed algorithm's effectiveness is demonstrated by displaying the classification results based on the load data of the service area of PG&E utility in the Western Electricity Coordinating Council (WECC). For the segmental dataset processor, 24 hours observation window dataset was used, and we classified all individual loads into four categories based on daytime and nighttime non-conforming feature. Because no label exists, a few examples for each type were presented, and we provided some suggestions and comments to the industrial community regarding each category. For the Full dataset processor, a one-month observation window dataset was used. We sorted all individual loads by their quantified non-conforming feature of the entire month and classified the non-conforming loads by a threshold from the sort. By referring to a label list subjectively generated by experts' experience in the control room, we analyze the impact of the hyper-parameter threshold on the classification performance and demonstrate the effectiveness of our approach. It shows that if we know the approximate proportion of non-conforming load in the entire data set, we will get a relatively good classification result. Fortunately, this is true, and this paper summarizes it as one of the three properties when considering non-conforming loads problems.

1.3 Regional Medium-Term Load Forecasting Problem

With the rapid development of human life, the complexity of power systems and the diversity of power consumers are increasing. In such an environment, the forecast of electricity demand/load has received extensive attention in the past few years [32]. People gradually realize that the forecast of power demand plays a significant role in the system planning and control of modern power systems, ranging from load allocation, generation planning to transmission augmentation [33]. This problem is challenging because its research needs to overcome a series of uncertainties. These uncertain factors can be divided into two categories from internal and external [34]. The internal factors which are firsthand related to the power system include the capacity of the existing power generation, the number of customers, the loading of transformers, and the flexibility of the behavior of customers; the external factors include the time change, the number of populations, climate change, and economic conditions, etc.

The load forecasting process can be divided into short-term load forecasting, medium-term load forecasting, and long-term load forecasting [33]. For short-term load forecasting, its lead-time usually from a few hours to a few days, and the primary objective usually focuses on high accuracy and fast prediction; for medium-term load forecasting, its lead-time generally from a few weeks to a few months, and the prior objective usually focuses on the perception of the peak load and hourly load profile; for long-term load forecasting, its lead-time usually cover few years to few decades, and the primary objective usually focuses on the perception of the peak load and the development trends. Even though some internal and external factors will influence the electricity demand process, these three categories of load demand forecasting need to consider factors differently. For example, the development of the power system, economic change, and population constantly change significantly influence long-term load forecasting. Still, the control room could ignore their impact on medium-term load forecasting because these factors hardly change obviously in a few months or years. In addition, climate change usually influences significantly to medium-term load forecasting, and time change is always essential for all kinds of load forecasting. Compared to the medium-term and long-term forecast, literature more focused on short-term forecast [35].

Before 2015, despite the load forecasting was essential for the expansion, operation, and planning of power systems, analysis of medium/long-term load forecasting has revived little attention [36]. At that time, the traditional regional or national electricity demand and its hourly profile were forecasted by scaling up the current electricity load [37–41]. However, after 2015, the growth electrification and customers' flexibility make the power-system planners have to take such changes into account to provide cost-optimal, and reasonable solutions [32]. Hence, medium/long-term load forecasting became the principal concern of both peak and hourly load profiles. With the development of technology in recent years, the methodology of solving load forecasting problems has three main trends: 1) the neural network-based approach, 2) the "big data" based approach, and 3) hybrid models.

Besides these high-level motivations mentioned above, the objective of medium/long-term load forecasting is diverse in practice and may be different in different regions or power grid environments. In recent years, some researches were proposed for various objectives by studying medium/long-term load forecasting with different methodologies. [35] worked on the density prediction and dimensionality reduction of the medium-term electricity demand based on a semiparametric-based additive model, which is a kind of hybrid model. [42] mainly focuses on studying the relationship between the temperature distribution and medium-term electricity demand forecasting based on the linear and non-linear regression approach. [43] explored the relationship between 24 hours for each day and 52 weeks for each year based on the short-term correlation and growth. The objective of [44] is the identification of consumption profiles and segmentation of customers based on long-term forecasting of hourly electricity load. [45] studied the different levels of wavelet for monthly medium-term load forecasting. There are also few pieces of research studied load forecasting for the various objective base on the Long-short term memory (LSTM): [46] took temperature, solar and a holiday into account to output the daily peak load for a long-term; [34] computed the yearly long-term load demand forecasting for Bali province using LSTM.

Except for the traditional motivation and existing objective mentioned above, a new motivation of medium-term load forecasting is proposed: serving for the real-time simulation. In the modern power system, State Estimator (SE), Real-Time Contingency Analysis (RTCA), and Automatic Generation Control (AGC) are primary Energy Management System (EMS) real-time applications widely used in control rooms. Compared to operation planning studies, SE, RTCA, and AGC are more dependent on real-time measurements on loads and generations to compute current system state variables, post-contingency power flows, and generation regulation control actions. Their existence provides necessary information and guarantees for the power system's planning, operation, and maintenance. However, the study of the upgrade for these applications faces challenges because of the extremely high data requirements, especially for the load data. To improve the resilience of power systems, these real-time applications need to simulate and study more extreme situations, which need enough load data to support them. Still, the data from the records usually lack sufficiency and flexibility. Therefore, high-resolution medium/long-term load forecasting is needed to provide efficient data support for the simulation of these real-time applications.

This research aims to generate or forecast the high-resolution (hourly) aggregated load curve for a region in the medium-term (a few months to a year). Firstly, we classify places with similar climatic characteristics into the same region because climatic (temperature, humidity, wind speed, etc.) highly influences people's daily lifestyle. Secondly, we decompose the region's hourly aggregated load curve into three parts: base-load term, seasonal-load term, and residual term. Then three parts are modeled and trained by different methods. Considering that almost all climatic or weather factors can ultimately be reduced to the temperature factor and time factor constantly influences load change, we take temperature and time factors into account. The base-load expresses the load curve's medium/long-term variation trend, and the input factor consists of daily average temperature and weekly time coding (to identify different weeks in a year). It is trained by a multi-layer perceptron (MLP). For seasonal-load, it expresses the seasonal or periodic load variation of both long- and short-term. The input factor includes the daily minimum to maximum regular temperature and weekday time coding (highlighting the difference between workdays and weekends, etc.). It is trained by Long-short term memory (LSTM). Finally, the residual term is treated as a time-series and was modeled by ARMA. By implementing the method in this research, a practical high-resolution aggregated medium-term load curve forecast can be obtained, which can not only realize the accurate prediction of peak load but also realize the depiction of hourly load profile and provide a reasonable and controllable load curve for the real-time simulation system as input.

To demonstrate the effectiveness of the proposed method, 2 case studies for California Independent System Operator (CAISO) and Bonneville Power Administration (BPA) load demand forecasting are presented. By the training of 2014 - 2018 load data, we predicted the 2019 load curve for CAISO and BPA. The results of case studies demonstrated that our approach has high prediction accuracy performance, and the time of the peak load's time for both CAISO and BPA are accurately predicted. Eventually, by analyzing the load forecasting results for the 2020 year of CAISO and BPA, we discussed the impact of the COVID-19 on people's daily lifestyles. The results show that COVID-19 changed people's daily lifestyle a lot, and it brings much more influence on CAISO than on BPA.

Chapter 2

Cyber-Attack Detection and Recovery

In this section, the spatial-temporal system model for the power grid is firstly introduced. Then the methodology to build the unknown kernel functions with the approximation of B-spline basis and to unknown parameters from the available training data is demonstrated. Finally, an anomaly detection scheme is provided based on the proposed system model, and some simulation results are presented as well.

2.1 The Spatial-Temporal System Model

In this section, we will build a spatial-temporal system model for the power grid. The power grid is a highly interconnected network of generators and loads connected by transmission lines. The nodes in this interconnected networks are usually termed as buses. Various models have been adopted to study the operations of power systems, e.g. the transmission line models, the models for the rotating machines and so on. However, with the development of power systems, traditional model-based approaches are now facing many challenges, e.g. the computational complexity introduced by the large number of elements and the non-linearity from renewable resources and complex load components. In this situation, data-driven approaches become a better alternative.

In the load flow study, which gives the basic representation of the operating state of the power systems, we are usually interested in four major quantities, namely the magnitude and the phase angle of the voltage, and the active and reactive power at each bus. These quantities can be measured by meters such as PMU installed on each bus as a function of time. The quantities across the entire system must follow some physical laws such as KCL, KVL, the conservation of energy, and the dynamic properties of the elements in the system. Therefore, there must exist an unknown relationship among the data measurements. Traditional model-based approaches try to capture the relationships among the various quantities by building up physical models for the elements and their connections. However, the model order can be really high for the power systems and any inaccuracy in the model of any individual elements could cause large error in the overall system model. To overcome these shortcomings, here we try to build a data-driven model to capture the underlying relationship among the voltages and currents in the power system both spatially and temporally.

We denote the variables (voltage and/or current) measured at bus i as x_i and try to represent x_i in terms of the variables at other buses. Denote d as the total amount of the buses in the power system, and the estimated value of each bus is denoted as $\tilde{x}_k(t)$ where $k \in \{1, \ldots, d\}$. Then, as illustrated in Figure 2.1, the model we try to build can be represented as follows:

$$\begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \vdots \\ \tilde{x}_d(t) \end{bmatrix} = \begin{bmatrix} f_1\left(x_2(t), x_3(t), ..., x_d(t)\right) \\ f_2\left(x_1(t), x_3(t), ..., x_d(t)\right) \\ \vdots \\ f_d\left(x_1(t), x_2(t), ..., x_{d-1}(t)\right) \end{bmatrix} \tag{2.1}$$

where f_i is the function to estimate $x_i(t)$ from $x_j(t)$'s where $j \neq i$ and the function is generally nonlinear.

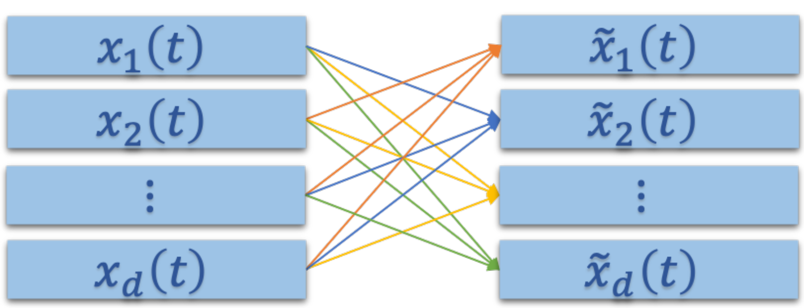

Figure 2.1: The system model. Assuming d is the total number of buses, the estimated value of each bus $\tilde{x}_i(t)$ on the right side is a function of the measured value of all remaining buses on the left side.

Notice that the power system is a dynamic system evolving in time and therefore, the variables in the functions $f_i(\cdot)$ are time series $x_i(t)$s. In order to handle the non-linear and the dynamic feature, this research adopts the Volterra series to build a data-driven model that captures the spatial-temporal relationships among these data measured at buses in the power system.

2.1.1 Volterra Series

The Volterra series is a nonlinear dynamical system identification model, which is also referred to as a Taylor series with memory [47]. Here, we explain the Volterra series with a simple example of just two variables. The model represents the variable of interest as the summation of different orders of convolutions of another variable, shown as follows:

$$x_k(t) = H_0 + \sum_{p=1}^{\infty} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} H_p(\tau_1, \dots \tau_P) \prod_{s=1}^{p} x_i(t - \tau_s) d\tau_s \ , \tag{2.2}$$

where in our context, $x_k(t)$ and $x_i(t)$ can be the data measured at two different buses k and i, and $H_p(\cdot)$'s are the unknown kernel functions for the p-th order relationship between $x_k(t)$ and $x_i(t)$. For $p > 1$, the kernel functions and the product across different variables capture the non-linearity. The convolution operations in the model capture the temporal dynamic relationships. This is also illustrated in Figure 2.2. The input for the model is the time series of the voltage at bus A, and the output is the time series of the voltage measured at bus B. The relationship is captured by the model and the kernel functions for different orders, $H_p(\cdot)$'s, which are usually unknown and need to be found based on the historical data.

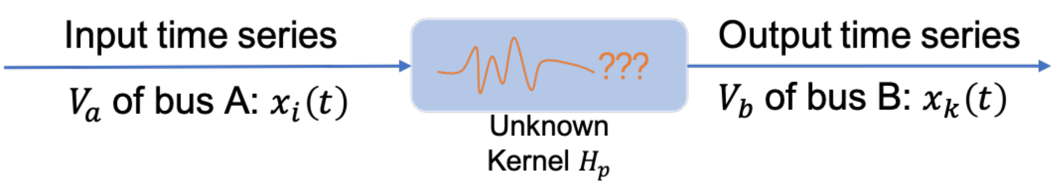

Figure 2.2: An example of the Volterra series in power system.

2.1.2 Spatial-Temporal Model

Using the Volterra series with $p = 2$ to model the $f_i(\cdot)$'s and including multiple input variables, a spatial-temporal model can be built [48] as follows:

$$
\begin{aligned}
\tilde{x}_k(t) =& h_0 + \sum_{i=1, i\neq k}^{d} \int_0^M \varphi_i^{(1)}(\tau) \cdot x_i(t-\tau) \cdot d\tau \\
&+ \sum_{i_1=1, i_1\neq k}^{d} \sum_{i_2=1, i_2\neq k}^{d} \int_0^M \int_0^M \varphi_{i_1 i_2}^{(2)}(\tau_1, \tau_2) \cdot x_{i_1}(t-\tau_1) \cdot x_{i_2}(t-\tau_1) \cdot d\tau_1 d\tau_2
\end{aligned}
\tag{2.3}
$$

Notice that different from the original Volterra series represented by (2.2), the integral is taken from 0 to M instead of from $-\infty$ to $+\infty$. It is because the power system is a causal system with finite memory and the system memory length is denoted to be M here. The Volterra order was chosen to be 2, in order to capture the non-linearity of power systems and the interactions in space and time. An order larger than 2 also could be used, but higher order will increase the computational complexity significantly and does not necessarily bring in much better fitting performance. Hence, (2.3) has three terms: the zeroth order term h_0 that captures the output state without any inputs, the first-order term that measures the impact of corresponding input data measured at buses, and the second-order term that measures the interaction effects between different time tags for the same input bus and between different inputs data measured at buses. Still, the $\varphi_i^{(1)}(\cdot)$ and $\varphi_{i_1 i_2}^{(2)}(\cdot)$ are some unknown kernel functions yet to be found.

2.2 Methodology

In this section, we introduce the methodology to build the unknown kernel functions with the approximation of B-spline basis and to estimate the unknown parameters from the available training data.

2.2.1 Approximation for the Unknown Kernel Functions

In order to parameterize the problem, the unknown kernel functions $\varphi_i^{(1)}(\cdot)$ and $\varphi_{i_1 i_2}^{(2)}(\cdot)$ are approximated by B-splines basis [49], similar to what [48] has done. The unknown kernel functions will be expressed as the linear combination of known basis functions at different knots:

$$\varphi_i^{(1)} \approx \sum_{j=1}^{J} \beta_j^{(i)} B_j(\tau) \tag{2.4}$$

$$\varphi_{i_1 i_2}^{(2)} \approx \sum_{j_1=1}^{J} \sum_{j_2=1}^{J} \beta_{j_1,j_2}^{(i_1,i_2)} B_{j_1,j_2}\left(\tau_1, \tau_2\right) \tag{2.5}$$

where J is the total amount of knots of B-splines basis. J is one of the hyper-parameters of our proposed model to be selected. In general, a higher J results in a higher approximation accuracy, but it will also bring the risk of over-fitting. One of the advantages of the B-splines approximation is that the kernel function have the flexibility to assign different weights on the input memory by customizing the distribution pattern of the B-splines knots. Here, without loss of generality, by using the equally spaced interior knots on the inputs memory, the entire kernel function will have the same weight.

2.2.2 Matrix Form of System Model

In order to facilitate the estimation of the system parameters, the system model is better written in matrix form. All known components including the B-spline basis functions and the inputs measurement value at buses, are assembled as $\xi_j^{(i)}(t)$ and $\xi_{j_1,j_2}^{(i_1,i_2)}(t)$.

$$\xi_j^{(i)}(t) = \int_0^M B_j(\tau) x_i(t-\tau) d(\tau) \tag{2.6}$$

$$\xi_{j_1,j_2}^{(i_1,i_2)}(t) = \int_0^M \int_0^M B_{j_1,j_2}(\tau_1,\tau_2) x_{i_1}(t-\tau_1) x_{i_2}(t-\tau_2) d\tau_1 d\tau_2 \tag{2.7}$$

Then, the system matrix could be defined as:

$$\boldsymbol{X}(t) = \left[\left\{ \xi_1^{(1)}(t), \xi_2^{(1)}, ..., \xi_J^{(d)}(t) \right\}, \left\{ \xi_{1,1}^{(1,1)}(t), \xi_{1,2}^{(1,1)}(t), ..., \xi_{J,J}^{(d,d)}(t) \right\} \right]^T \tag{2.8}$$

And all unknown components system parameters $\beta_k^{(i)}$ and $\beta_{k_1,k_2}^{(i_1,i_2)}$ are organized into another column vector:

$$\boldsymbol{\beta} = \left[\left\{ \beta_1^{(1)}, \beta_2^{(1)}, ..., \beta_J^{(d)} \right\}, \left\{ \beta_{1,1}^{(1,1)}, \beta_{1,2}^{(1,1)}, ..., \beta_{J,J}^{(d,d)} \right\} \right]^T \tag{2.9}$$

Then, the system model can be written as:

$$\tilde{x}_k(t) = h_0 + \boldsymbol{\beta}^T \boldsymbol{X}(t) \tag{2.10}$$

It should be noted that (2.10) is a linear model with respect to $\boldsymbol{\beta}$, but the model itself is nonlinear of $x_i(t)$'s since the elements in $\boldsymbol{X}(t)$ results from nonlinear operations on $x_i(t)$.

2.2.3 Estimation of Unknown Parameters

When some historical data are available, they can be used to train the model represented in (2.10). Solving the coefficient $\boldsymbol{\beta}$ in (2.10) is a well-done subject in the statistical learning literature, and many estimation algorithms can be applied such as Minimum Mean Square Error (MMSE) estimation [50], Maximum Likelihood Estimation (MLE) [51] and Kalman Filter estimation [52]. Given that in power systems the accurate distribution and the prior information for the additive noise of measurement are usually unknown, we choose to apply the least square estimation.

However, singularity issue often occurs when solving the model. It mainly comes from two reasons: 1) the local similarity of the original power system measurement data; 2) the local similarity of high interaction by the 2nd-order term in the proposed system model. These two similarities will cause high correlation between some rows and columns in the system matrix $\boldsymbol{X}(t)$ in (2.10), which in turn leads to an approximately singular state of the matrix. It will make some elements in the least square solution relatively large and then causes a large estimation error. Therefore, the damped least square estimation is used [53]:

$$\hat{\beta}_{DLS} = \arg\min_{\beta} \left(\left\| x_k(t) - h_0 - \boldsymbol{\beta} \cdot \boldsymbol{X}(t) \right\| + \lambda \cdot \sum |\beta_i|^2 \right) \tag{2.11}$$

$$\hat{\beta}_{DLS} = (\boldsymbol{X}^T \boldsymbol{X} + \lambda \boldsymbol{I})^{-1} \boldsymbol{X}^T (x_k(t) - h_0) \tag{2.12}$$

where $\boldsymbol{I}$ is the identity matrix. The added damping item $\lambda \cdot \sum |\beta_i|^2$ in (2.11) will minimize not only the distance between the estimation and measurement of (2.10) but also the L-2 norm of coefficient vector $|\beta_i|^2$, which can prevent $\boldsymbol{\beta}$ from generating extreme values. The damping degree is affected by the scalar λ. In general, the stronger the singularity of the system matrix, the larger λ is needed.

2.3 Anomaly Detection

Our proposed spatial-temporal system model has captured the relationship between the measurement at each bus as the function of the measurement at all other buses under normal system operating conditions. Therefore, we can conduct anomaly detection by checking the difference between the true measurement and the measurement estimated from our model. Once the residual between the estimated and the true measured value is abnormally large, we can claim that anomaly happens in the measurement. Either the measurement at the bus itself is abnormal, or the measurement at one or more other buses which serve as inputs to our model is abnormal. To conduct anomaly detection based upon the residual, we first find the distribution of the residual under normal conditions and then formalize the detection scheme.

2.3.1 The Distribution of the Residual under Normal Conditions

The residual vector $\boldsymbol{e}(t)$ is defined as the collection of errors between the true measured value and the estimated value from our proposed spatial-temporal model at all buses:

$$\boldsymbol{e}(t) = \begin{bmatrix} e_1(t) \\ e_2(t) \\ \vdots \\ e_d(t) \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_d(t) \end{bmatrix} - \begin{bmatrix} \tilde{x}_1(t) \\ \tilde{x}_2(t) \\ \vdots \\ \tilde{x}_d(t) \end{bmatrix} \tag{2.13}$$

where d is the total number of buses under study in the system. In general, the residual should be composed of two parts: 1) the measurement noise and 2) the mismatch between our proposed model and the true underlying relationship between quantities. The measurement noise is usually additive zero-mean Gaussian noise. The model mismatch noise can also be assumed to be zero-mean Gaussian distributed given that our estimation process is unbiased and usually involves large number of training data. In addition, we can assume that different rows of the residual are independent since the spatial-temporal models for the measurement at different buses are actually trained independently. As a result, we can assume that i-th row of the residual each follow a Gaussian distribution with zero mean and variance σ_i^2, denoted as $e_i(t) \sim N(0, \sigma_i^2)$. The variance σ_i^2 is unknown and need to be estimated from our training process.

For the training dataset, at each bus i, one can generate the estimated value from the resultant trained spatial-temporal model, and then compute the residual e_i and estimate the variance σ_i^2 based upon the training data. Here, the maximum likelihood estimation is adopted [51]. This is the optimal estimate and is easy to obtain under the Gaussian distribution assumption.

For the training dataset, at each bus i, one can generate the estimated value from the resultant trained spatial-temporal model, and then compute the residual e_i and estimate the variance σ_i^2 based upon the training data. Here, the maximum likelihood estimation is adopted [51]. This is the optimal estimate and is easy to obtain under the Gaussian distribution assumption.

The likelihood function of σ_i^2 could be written as follows:

$$L\left(e_i(t)|\sigma_i^2\right) = \prod_{t=1}^{N} \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{e_i(t)^2}{2\sigma_i^2}\right) , \tag{2.14}$$

where N is the total amount of training data. In order to facilitate the optimization problem, the natural logarithm is taken on the both side of (2.14) to convert the production into summation. The log-likelihood function could be written as:

$$\ln L\left(e_i(t)|\sigma_i^2\right) = \sum_{t=1}^{N} \left[-\frac{e_i(t)^2}{2\sigma_i^2} - \ln\left(\sqrt{2\pi\sigma_i^2}\right)\right] . \tag{2.15}$$

Then the estimate for the variance σ_i^2 could be obtained by maximizing the log-likelihood function with respect to σ_i^2:

$$\hat{\sigma}_i^2 = \arg\max_{\sigma_i^2} \ln L\left(e_i(t)|\sigma_i^2\right) . \tag{2.16}$$

2.3.2 Anomaly Detection

With the estimated σ_i^2 and the Gaussian assumption, the probability density function (PDF) of residual for the i-th bus under normal conditions (or null hypothesis) can be written as:

$$f\left(e_i(t)\right) = \frac{1}{\sqrt{2\pi\hat{\sigma}_i^2}} \exp\left(-\frac{e_i(t)^2}{2\hat{\sigma}_i^2}\right) , \tag{2.17}$$

The distribution of the residual under abnormal conditions would be unknown since there could be various kinds of abnormal conditions. Therefore, here we propose to detect the anomaly solely based upon the distribution under normal conditions. For the given distribution, the anomaly detection can be simply set as a one-sided χ^2 test [54]:

$$\frac{|e_i(t)|^2}{\hat{\sigma}_i^2} \geq \chi^2_{2,P_f} \ , \tag{2.18}$$

where χ^2_{2,P_f} denotes the critical value of the χ^2 distribution with 2 degrees of freedom with confidence level P_f and P_f is the false alarm (declaring anomaly while it is actually normal) probability for the detection.

In the context of cyber-attack, for the Denial of Service (DoS) attack, the measured value at attacked buses will be unavailable (equal to zero) when DoS occurs, but the estimated value can be still obtained via our proposed model. Therefore, large residuals for the buses under attack occur, and then the anomaly will be identified. For the False Data Injection (FDI) attack, the measured value of the attacked buses will be replaced by false data, but the replaced false data will be in disagreement with the modeled spatial-temporal relationship. Therefore, large residuals for the attacked buses occur, and then the anomaly will also be identified.

2.4 Simulation Results

Our approach was tested with the data from the minniWECC simulations [55] which is based upon a simplified version of the system monitored by the Western Electricity Coordinating Council (WECC) of North America. The minniWECC system model includes 120 buses, 171 lines, 34 generators, 20 load transformers, 19 load buses and 2 DC lines. The simulation will generate PMU data for each Bus and Line, and the sampling rate was set up as 5 samples per second. In all simulations, the voltage data of the PMUs were used. The current data were not used here since there is no obvious difference in performance between using voltage data and current data. In practice, if one kind of data of some buses are missing, people can replace it with another kind of data. In other words, our model would be applicable to any kind of combinations of data. We trained our model with 1 hour data under normal operating conditions, which means that the power system has no major events and no attacks except for the 20% random load variation.

In this section we present three case studies. The first case study shows the performance of our proposed data-driven spatial-temporal model to fit the power system data, and the impact of the hyper-parameters in our model on the fitness performance. The second case study shows the recovery performance when the DoS attack happens at a single bus and explains the impact of the number of attacked buses on the average recovery performance. The third case study shows the detection performance when FDI attack occurs by trying to inject some historical data to replace the true measurement.

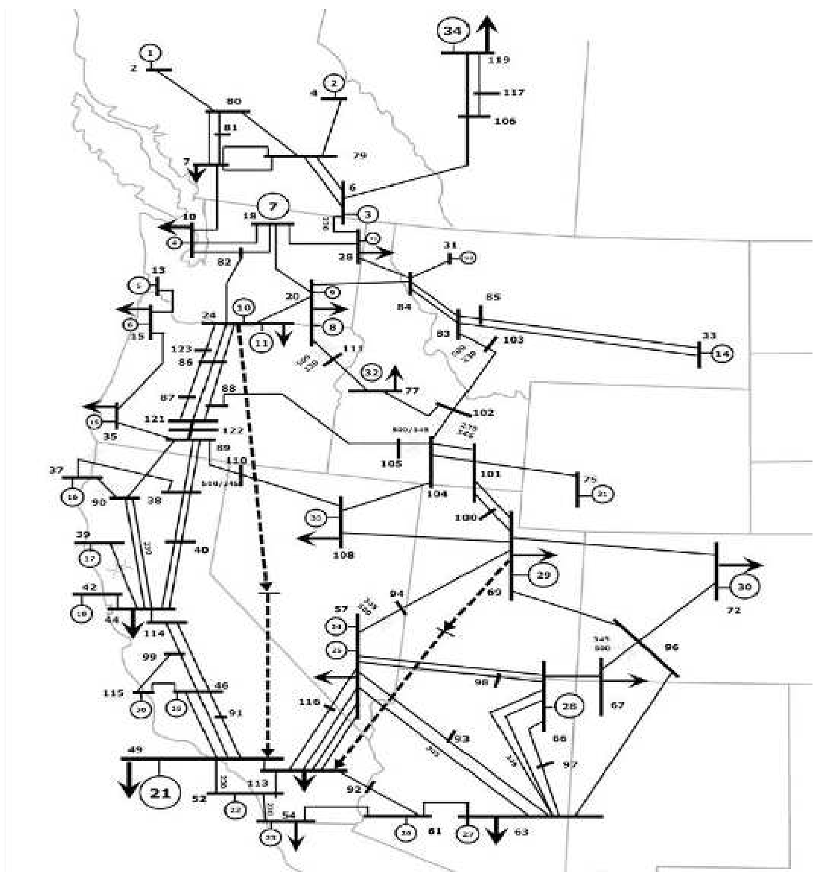

Figure 2.3: The minniWECC simulation system.

2.4.1 Case A: The Goodness of Fit of Our Proposed Model

To show the goodness of fit on power systems, we have randomly select 9 buses in the minni-WECC system, and simulated data to be used as the training set and validation set with one hour gap between them. In this case, both the training dataset and the validation dataset are the power system ambient data under normal operations, to demonstrate the goodness of fit of our proposed data-driven spatial-temporal model.

Table 2.1: Evaluation

	Training Set			Validation Set		
Bus Num.	RMSE	Error(‰)	R^2	RMSE	Error(‰)	R^2
8	0.0051	0.0081	0.8784	0.0059	0.0092	0.8279
10	0.0044	0.0066	0.9742	0.0048	0.0072	0.9730
16	0.0079	0.0273	0.3861	0.0088	0.0301	0.3022
21	0.0061	0.0098	0.6922	0.0072	0.0114	0.5633
27	0.0031	0.0232	0.4672	0.0035	0.0259	0.2739
35	0.0069	0.0111	0.9383	0.0075	0.0120	0.9243
43	0.0057	0.0089	0.9423	0.0064	0.0098	0.9358
75	0.0001	0.0033	0.9667	0.0001	0.0037	0.9633
111	0.0003	0.0119	0.9009	0.0004	0.0147	0.8682

Table 2.1 shows the fitting performance of our model on the selected 9 buses (Buses 8, 10, 16, 21, 27, 35, 43, 75 and 111 in the minniWECC system) and works on the voltage measurement on each of these buses with all other remaining buses' voltage as the inputs. We use the Root Mean Square Error (RMSE) as the main performance metric. The "Error (‰)" means the average relative error, which is calculated by the average of $\left|\tilde{x}_i(t) - x_i(t)\right| / x_i(t)$ for $t = 1, 2, \ldots, n$ with the unit of per thousand. Also, we use the R^2 test to evaluate the degree of similarity between the two curves and the value is between 0 to 1 [56]. The higher the R^2 value, the better the fitness of the model on the data.

From the results, we can see that the RMSE and the relative error for each chosen bus are very small, and the R^2 test scores for them are close to 1 except for the No. 3 and No. 5 buses.

These two buses have R^2 test score lower than 0.5 but have a fair accuracy for the RMSE and the relative error. The reason for this is that the measurement of these two buses has a relatively high non-linear relationship that is captured by our model, but the R^2 test mainly quantifies the degree of any linear correlation between the observation and the estimation. This result shows that the proposed model has good fitting performance on power system data.

The proposed model has three hyper-parameters which need to be chosen by users for the different applications. In order to demonstrate their impact on the fitness performance, we conducted 300 random tests on each of them separately and show the results in the form of box plots.

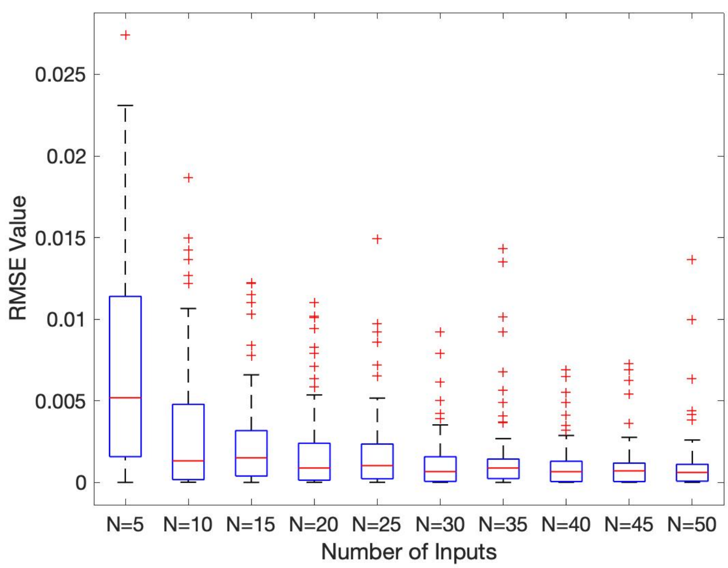

Figure 2.4: The test on the number of inputs.

Firstly, this research tested the influence of the number of inputs N with different values from 5 to 50. And for each N, we randomly selected nodes with the corresponding number of inputs from the minniWECC simulation system 300 times. Figure 2.4 shows that with the increase in the number of inputs our model will provide more and more accurate performance. On each box in the figure, the central red mark indicates the median, and the bottom and top edges of the box indicate the $25th$ and $75th$ percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers and the outliers are plotted individually using the

'+' symbol. Same as the test above, the voltage of the corresponding buses are used as inputs. Considering the computation cost ratio, the number of inputs as 20 was chosen in the rest of the simulations.

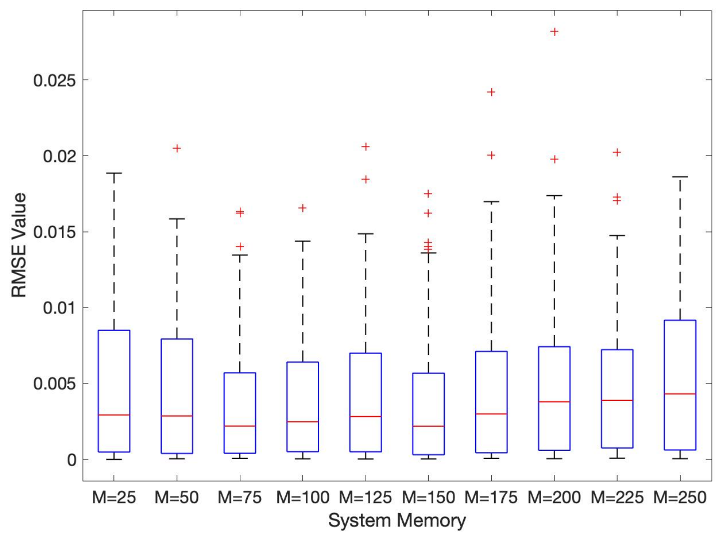

Figure 2.5: The test on the system memory.

Secondly, the system memory was tested with different value from 25 to 250, and we also executed 300 tests for each system memory with randomly picked 20 buses. This system memory affects how many temporal information that our model will include. From Figure 2.5 we can know that a small system memory will make our method in lack of sufficient temporal information to obtain the best performance, in other words, resulting in the under-fitting situation. A big system memory will make our method obtain too much redundant and unrelated temporal information, which will reduce the efficiency and accuracy of the algorithm, or equivalently ending up in the over-fitting situation. For the minniWECC system under study we obtained the best performance at $M = 75$ and this research will fix it for the last test of hyper-parameter.

Then, this research tested the last hyper-parameter the number of knots J, with also 300 random tests. In our simulation environment, we use the uniformly distributed knots, so the only factor matters is the total amount of knots. It will affect the density of the basis functions dis-

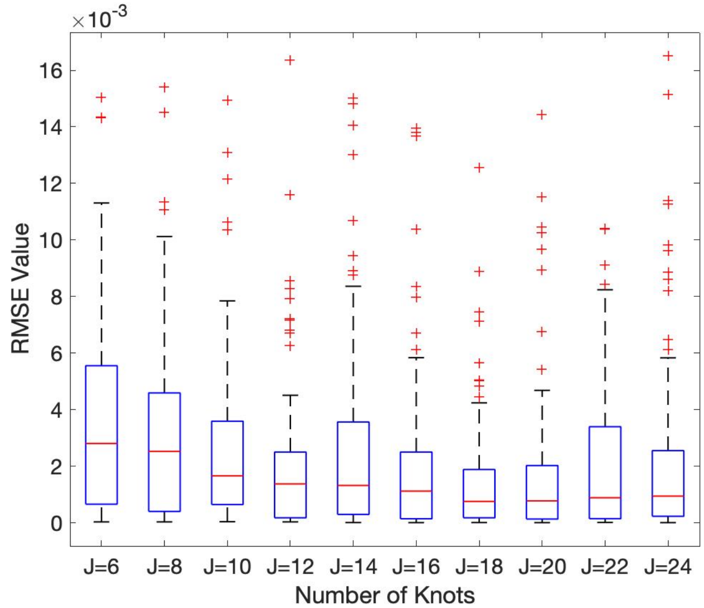

Figure 2.6: The test on the number of knots.

tributing in the time period of the system memory during the B-splines approximation process. Figure 2.6 shows the fitting performance under different number of knots. Same as the other two hyper-parameters, a different number of knots should be selected for different power systems. A smaller number of knots may not be enough to satisfy the higher complexity of the unknown kernel function, and a higher number of knots will result in a very dense basis function, and therefore may be too aggressive to meet the characteristics of the unknown kernel function, resulting in over-fitting. Actually, users can customize how these knots are distributed based on prior knowledge of the general shape of the kernel function or based on the user's subjective interests.

2.4.2 Case B: The Application of Our Proposed Model to DoS Attacks

Based on the high accuracy of our system model in fitting the power system data, we can effectively combat against the Denial-of-Service attacks (DoS). A DoS attack will block the measurement feedback function of a certain bus or some buses, resulting in the loss of the measurement data of the bus, and thereby makes the real-time monitoring mechanism of the control center in untrusted state. In this section, this research will discuss the performance of our model under both single-bus attack and multiple-bus attack situation.

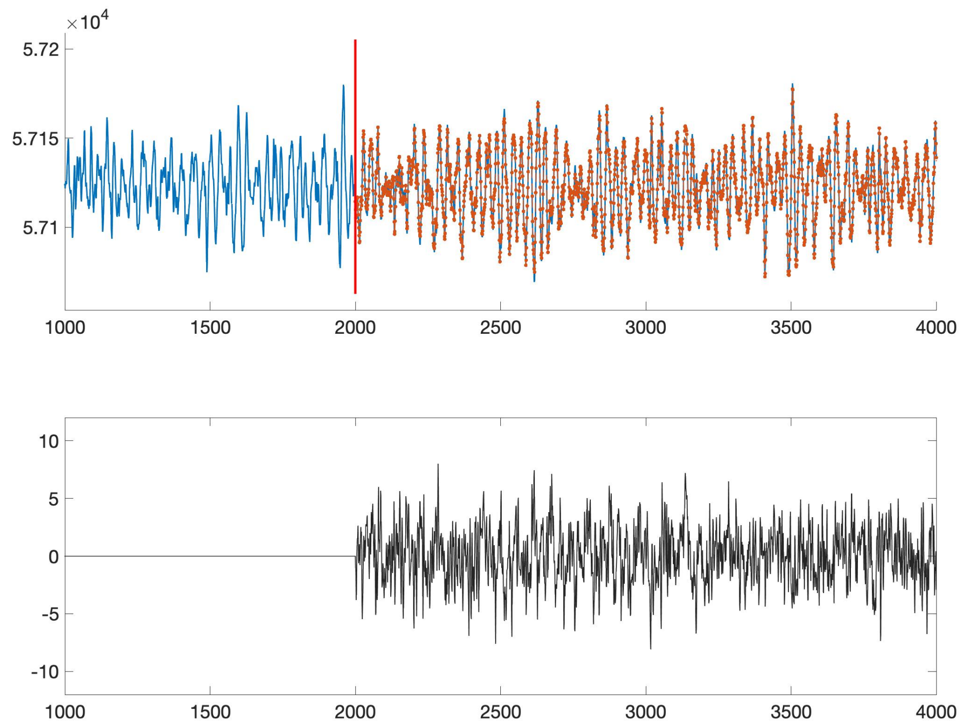

Figure 2.7: The test of the Number of buses. In the top figure, blue curve is the PMU measurement data of one bus, and the orange curve with dots is the predicted value provided by our method after the DoS attack occurs at $t = 2000$. In the bottom figure, the black curve is the residual between the measurement and the predicted voltage value at bus.

When the DoS attack occurs, the attacked bus cannot obtain any measurement. In this case, we can use our proposed spatial-temporal model as a predictor to generate the estimated data from other un-attacked buses and provide them to the control center as a substitute for the lost data due to the attack. For example, in Figure 2.7, a DoS attack occurs at $t = 2000$ on the PMU measurement data of one bus. The blue curve of the top plot of Figure 2.7 is the current magnitude measurement value of the attacked bus. After the DoS attack occurs, our method started working and provide the predicted value in real time (seen as the orange curve with dots). The top plot is showing that the orange curve (estimated value provided by our model as substitute) and the blue curve (original measurement if no DoS attack happens) are very close, and the bottom plot of Figure 2.7 is the residual $\boldsymbol{e}(t)$ from equation (2.13). The residual before $t = 2000$ is equal to zero because the DoS attack did not happened yet. We can see that the residual value in the bottom plot is four magnitudes of order smaller as compared with the measurement value in the top plot. We also checked the autocorrelation of the residual $\boldsymbol{e}(t)$ which is approximately equal to 0.197, which means that $\boldsymbol{e}(t)$ is close to white noise.

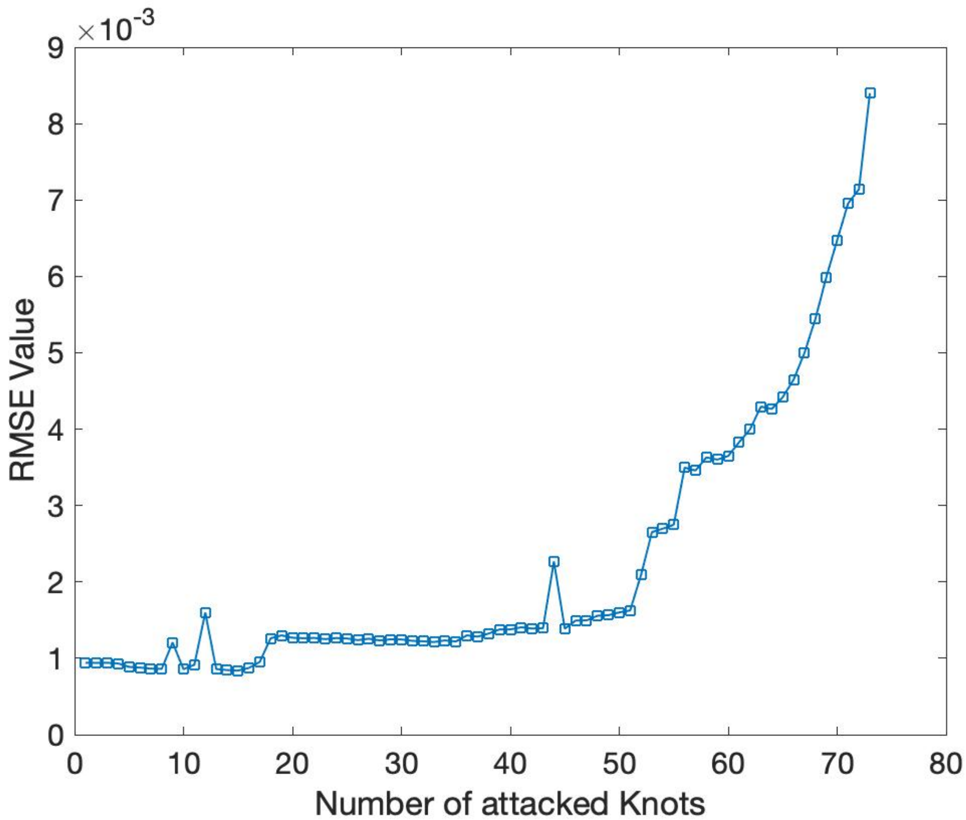

Figure 2.8: As the number of attacked buses increases, the average absolute error of the calculated values of all attacked buses increases.

In practice, attackers might carry out DoS attacks on multiple buses at the same time instead of just targeting at one bus. In order to test the performance of our method in this article when multiple buses are attacked by DoS, we incorporate all 121 buses in the minniWECC simulation system as the inputs of our proposed model and monitor the measurement at them in real time. Then we test cases where the attacker randomly selects the attacked buses and gradually increase the number of attacked buses. In each group of test, we use the remaining secure buses (unattacked buses) to calculate the estimated value of each attacked buses, and each group performs 100 random tests and calculates the average absolute error (RMSE) of the estimated value of all attacked buses. The average voltage value of the entire 121 buses is about 290.07 KV, hence the normalized RMSE still remains at the level of 3.447×10^{-5} after 70 out of 121 busses were attacked. Finally, the largest number of attacked buses in our test is 73, leaving only 48 unattended buses. Figure 2.8 shows the average absolute error (RMSE) versus the number of attacked buses. From the results, we can see that when the number of attacked buses is less than 51, the average absolute error (RMSE) of the estimated value of all attacked nodes in the system is relatively small and the growth rate is relatively slow, which means that the performance of our algorithm, in this case, is good and relatively stable; when the number of attacked buses is greater than 51, the average absolute error (RMSE) of the estimated value of all attacked buses in the system begins to increase rapidly, which means that the performance of our algorithm begins to decline in this case. This is quite reasonable since most measurement in the system become unavailable and very limited information can be provided to our model. Nevertheless, even under this extremely bad situation, our algorithm still has a small error when $2/3$ of the buses of the system are subjected to DoS attacks.

2.4.3 Case C: The Application of Our Proposed Model to FDI Attacks

In addition to the threat of DoS attacks, the power grid in recent years is also being threatened by False Data Injection (FDI) attacks, which are mainly conducted through intrusion into the communication network of the measurement system to change or inject the measured values of the sensors in the power system. One of the most threatening is the stealthy attack, which is proven to bypass the Bad Data Detection (BDD) module for current power system monitoring and control center, and then guide the control center to obtain wrong state estimate.

An example is shown in Figure 2.9. When a bus is attacked by FDI from $t = 1550$ to $t = 2550$, the attack is detected immediately through the sudden increase of residual.

Many studies have proposed various detection or recovery solutions for stealthy attacks relying on the difference between the bad data and the original data (see e.g. [23–25]). However, if the attack data itself is the true data collected from the system but only at different times, these meth-

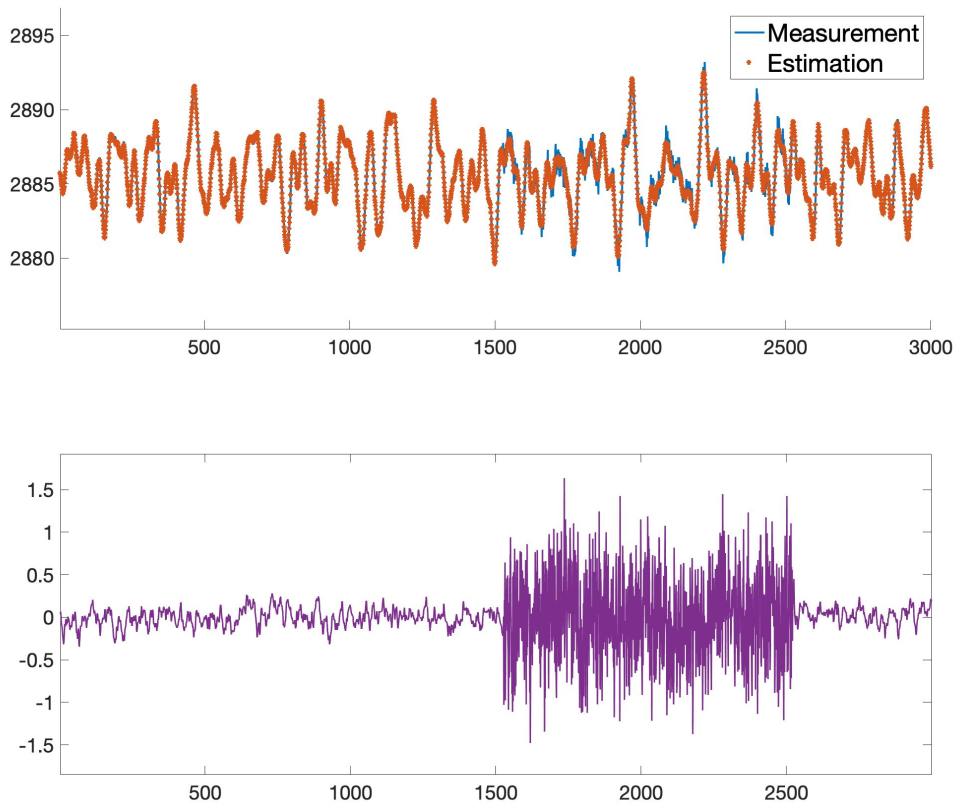

Figure 2.9: A historical data FDI attack. Some historical data were injected to measurement curve from $t = 1550$ to $t = 2550$. The bottom figure is the residual between the measurement and the predicted/estimated value.

ods are often not effective since most of them make the detection solely based upon the spatial relationship among the data while neglecting their temporal relationship. Here, we bring attention to this new type of attack, namely the historical normal data injection attack. We can see that our proposed method can not only effectively detect this type of attack, but also effectively recover the original data from our spatial-temporal model. The reason is that our proposed method captures both temporal and spatial information and therefore even if some historical data is used to replace the true measured data, they can still be identified as wrong data because they would not maintain the temporal relationship of the true system operating data.

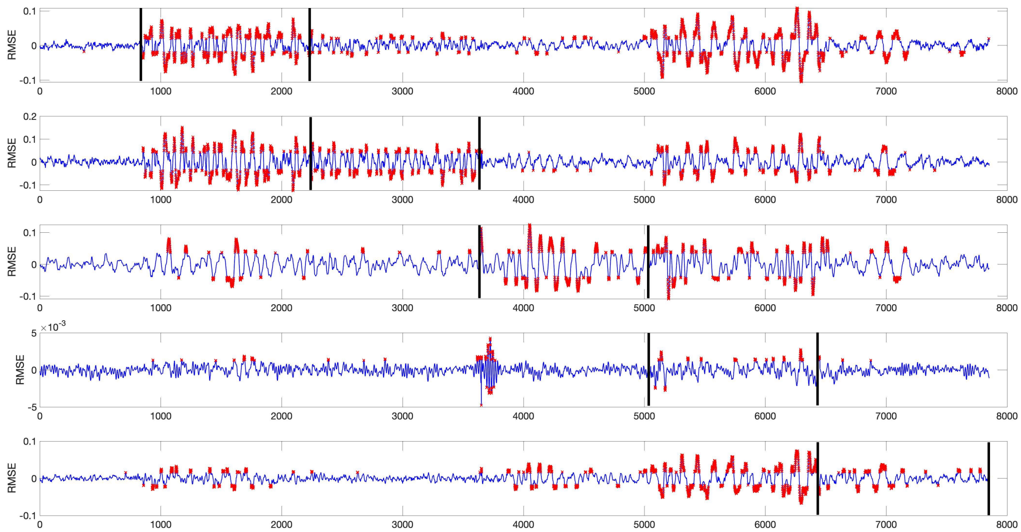

Figure 2.10: The historical normal data attack on 5 randomly selected buses. The intervals marked by two vertical black lines are the attacked periods, and the blue curves are the residuals between measurement and predicted value. The red dots denotes the abnormal samples detected by our method.

We randomly select 5 buses in the minniWECC simulation system and use the historical data from their training set to inject them in sequence. Each time only one bus was attacked. In Figure 2.10, the same duration interval marked by two vertical black lines on each bus means the attacked time period. For example, the first bus was attacked from $t = 800$ to $t = 2200$, and the second bus was attacked from $t = 2200$ to $t = 3600$, and so on. The red dots denotes the abnormal samples detected by our anomaly detection method.

Figure 2.10 shows that before $t = 800$ the residual of 5 buses are very small and the system behaves normally, but when the historical data was injected, the residual of 5 buses begin to increase and exceed the anomaly detection threshold and were immediately detected as abnormal samples. This phenomenon illustrates that in real-time running one bus attacked by FDI will affect the residuals of other buses at the same time. This is because that any bus that is attacked will become the input of the predicted value of the other buses at the same time and we do not know which buses can be trusted. Therefore, our method can just detect the FDI attack but cannot recover it fully at the present stage. More work need to be done for the recovery of power systems data attacked by FDI attack with historical data.

2.5 Summary

In this chapter, the detection and recovery of availability attack and integrity attack in cybersecurity of the smart grid are investigated and a new attack type of FDI named historical normal data injection was considered. To improve the accuracy of fitting the PMU data of power systems, this research models the spatial-temporal relationship among data collected across the entire system. This research demonstrates the applicability of the proposed method in this research through the simulations of ambient data of the minniWECC system, and the fitting accuracy can explain the recovery capability of the proposed method to the DoS attacked bus, using the data from remaining secure buses. By simulating the DoS attack, this research demonstrated the detection and recovery performance of the proposed method when a DoS attack occurs on a single bus and explains the impact of the number of attacked buses on the overall performance of the proposed method. This research uses historical normal data as false data to simulate the FDI attack and demonstrates the detection performance when FDI occurs on buses of a power system.

Chapter 3

Non-Conforming Load Classification

In this section, the details of the importance of the non-conforming load is firstly provided. Then, the methodology of the two classification processors is introduced. Some experimental results are presented at the end of this section.

3.1 Non-Conforming Load in Power Systems

As we mentioned in the Section 1.2, the non-conforming load was defined as the kind of load with the characteristics of irregular changes, which is the opposite of the residential load (conforming load) such as the commercial, industrial and agricultural load. In the past, when the types and quantities of loads in the power system were small, the demand for automatic classification of individual loads was not so urgent. At that time, utilities used holistic and statistical approaches to determine whether the individual load was non-conforming. Meanwhile, the changes in load characteristics had been prolonged and gradual due to the absence of Distributed Energy Resource (DER). Once a year or several years, a review was proven adequate to determine individual non-conforming loads and model them in the energy management system (EMS) as simple base value loads.

With the gradual formation of smart grids, the problem of non-conforming load classification has become more severe [57]. The large amount of load data collected by the Wide-Area Measurement System (WAMS) makes the control center's perception of the load in the power system fuzzy [58]. The complex structure and flexible form of these loads have brought new challenges to the classification of non-conforming loads [59]. In addition, because of the growth of residential loads with rooftop solar panels (behind the meter), prone-nonlinear power electronic loads, and DER, the definition of the load has lost its original in-habitability. Also, because of the residential renewable generation and power electronic load in emerging power grids, the load characteristics change more quickly than ever. In the power system modeling, both generator and load exist on a bus, so power injection at the bus itself has duality and varies significantly over time. For example, in some buses combined with renewable energy sources such as solar PV, the injection on the bus during the day will be dominated by solar generation output-positive. At night, the injection on the bus will be negative and supplied by bulk system generations, and it will behave as a pure load. Such a new form of the load is labeled "efficient load" in western utilities, such as PG&E and Idaho Power Company. These new features and changes make the existing non-conforming load identification methods no longer adequate to handle new challenges.

Non-conforming load impacts the operation planning study results directly. Given a grid system like CAISO (California Independent Operator System), which includes thousands of efficient loads, it results in noticeable errors for engineers to perform future operation planning studies by existing load model approaches to represent efficient loads. With the growth of renewable resources, the accuracy of the planning study that uses these efficient loads will depend on the different periods. For example, if it is a sunny day, the load profiles of the planning study may be accurate, but if in the evening, the load profiles will be different. A good perception of the non-conforming load will benefit the accuracy of the system planning study.

Non-conforming load also impacts the primary EMS real-time applications: State Estimator (SE), Real-Time Contingency Analysis (RTCA), and Automatic Generation Control (AGC), which are widely used in the control rooms. Compared to operation planning studies, SE, RTCA, and AGC are more dependent on real-time measurements on loads and generations to compute current system state variables, post-contingency power flows, and generation regulation control actions. Typically, EMS only receives telemetry values for only 50% of individual loads while handling the rest of non-measured loads as pre-modeled pseudo power injections. Estimation of non-telemetry loads heavily dependent on normalization of pre-modeled pseudo power injections across the system or area called load allocation factors. A conforming load is assumed to follow system load variation via a constant load allocation factor. For a non-conforming load, the load allocation factor is not fixed. Identifying non-conforming load and modeling its allocation factor with variable values is essential to obtain accurate solutions of SE, RTCA, and AGC.

From the past to the modern power system, the non-conforming load analysis plays an increasingly important role. However, few researchers in academia have contributed to this field because the research on this issue requires sufficient real data as support, and data resources in the past were very scarce. Until later, both EMS and Distribution Management System (DMS) have been operating for decades, and each utility control room has collected a large amount of load estimation data. The resolution of these data ranges from 1-5 minutes intervals in SE to 10-second intervals of ICCP measurements. The rich data and information should be good enough for using new technologies for automatic classification or identification of individual loads. The new approach will enable utilities to accurately manage load estimation in real-time operation and planning horizons to meet emerging challenges and needs driven by the "green energy" wave.

In the experimental environment of this study, to explain the algorithm applied in this research, this research uses a data set as an example in the rest of the content. The data set consists of 1610 individual loads in the service area of PG&E in WECC. Each individual load records the load data for one month at a sampling interval of 5 minutes. The following section demonstrates the methodologies separately based on the different focus of periods to interpret the differences and characteristics of the two proposed approaches.

3.2 Methodology

The power grid can cover a wide range, and it can be divided into different sub-regions according to factors such as geography or climate. Customers in such the same sub-region usually have similar characteristics to some extent. In this paper, we aim to classify the individual loads in a sub-region of power system into two categories: conforming or non-conforming load, based on the difference between the features of each individual load and the features of the aggregated load of the sub-region. One of the difficulties of this research is that there is no clear definition of non-conforming load. In other words, the data is not labeled, so our algorithm uses an unsupervised classification idea. For unsupervised classification problems, clustering is the core of the algorithm and the basis for further classification. Appropriate feature extraction is a prerequisite for clustering analysis. To perform better feature extraction, the original data set should perform data pre-processing, dimension reduction, and feature dissociation. Then according to the non-conforming extent of individual loads may change over time. This paper proposes two classification algorithms for segmental and full dataset processors based on different feature extraction methods using various clustering methods. A flow chart of the proposed algorithm is shown in Figure 3.1.

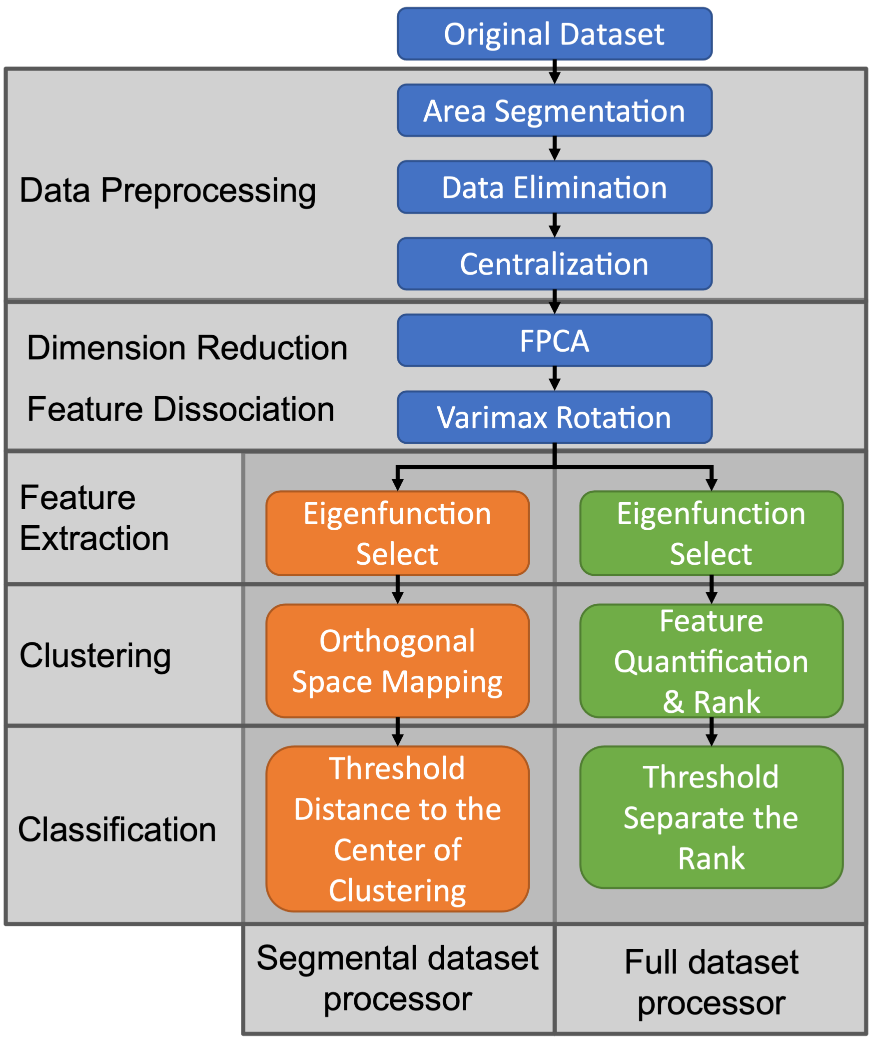

Figure 3.1: Algorithm flow chart.

To solve the classification problem for non-conforming loads, the three properties have to be considered:

1) *Regularity*: the conforming loads have a similar regular and seasonal pattern.

2) *Majority*: there is a large number of individual loads that are conforming.

3) *Proportion*: the experts in the WECC have a rough idea for the total amount of the non-conforming loads in an area.

The three properties are the critical basis for feature extraction, clustering, and classification of non-conforming loads. Next, this section will introduce the data preprocessing process and use the above three properties to achieve the final classification purpose.

3.2.1 Data Preprocessing

The data preprocessing is divided into three steps: area segmentation, data elimination, and centralization.

Area Segmentation:

When a power grid spans a large area, such as WECC, loads in different areas may have essentially different performances due to natural factors or other external factors. For example, in California, the peak load of an entire year will occur in summer because people use air conditioners to keep cooling. Still, in Seattle, the peak load of a whole year will occur in winter because people use heaters to keep warm. In this case, it is unfair to compare the characteristics of an individual load with the aggregated load of the entire grid. Therefore, it is necessary to divide the vast area into several sub-regions according to factors such as humanities or geography. Here we take WECC as an example, according to the climate zones provided by [60] to divide the entire Western Interconnection (WI) footprint into 12 sub-regions to illustrate this methodology, as shown in Figure 3.2:

The original dataset can be presented as:

$$W = A_1 \cup A_2 \cup \cdots \cup A_{12} \tag{3.1}$$

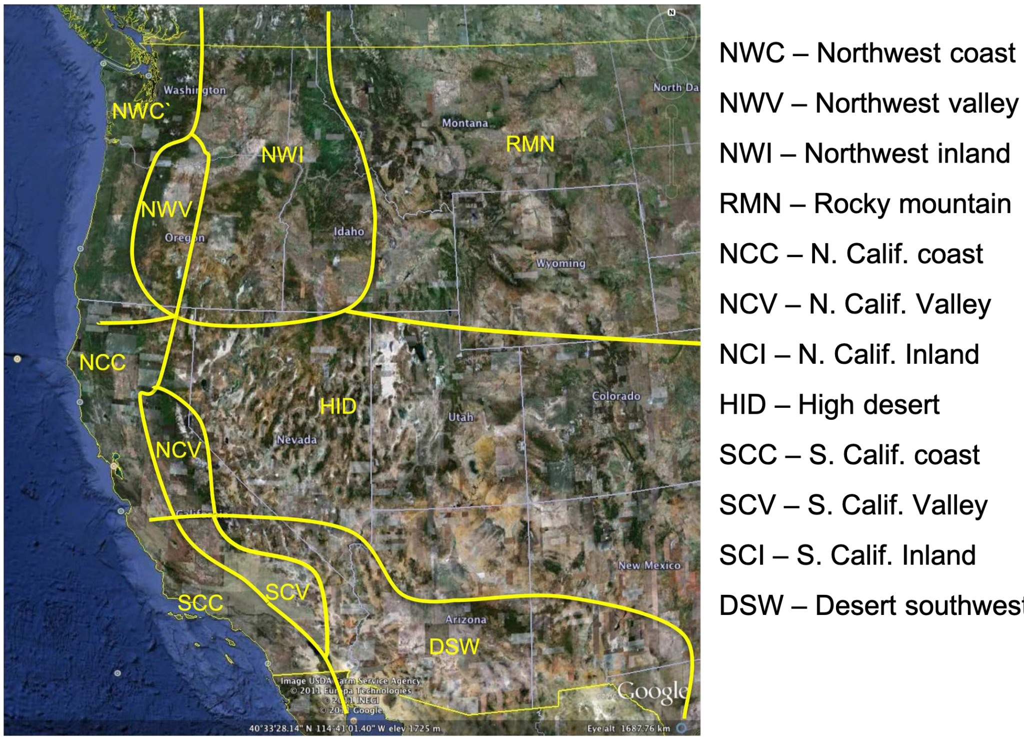

Figure 3.2: Climate zones.

where W is the set including all individual loads in WECC, and A_i, $i = 1, 2, \ldots, 12$, are the 12 sub-regions divided by climate and temperature because they are the most two important factors that influence the customers.

Data Elimination:

For any sub-region $A_i, i \in 1, 2, \ldots, 12$, not all of the individual loads in it must be considered. For the control center, those loads with larger watts are often more concerned, and The control room can usually ignore those with smaller watts because they have almost no effect on the system's operation. However, in our subsequent calculation process, the mean value of the individual load is not considered. In other words, a load with a more considerable watt value and a load with a small watt value have the same effect in the algorithm, which we hope to avoid. Therefore, the elimination of the dataset is necessary.

For the i-th sub-region A_i, we assume it includes k individual loads. The load characterization is based on time series analysis. Therefore we assume that each individual load record has an observation window with length N, which makes each sub-region dataset to be organized into a $k \times N$ matrix:

$$A_i = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \\ \vdots \\ \mathbf{x}_k(t) \end{bmatrix} = \begin{bmatrix} l_1^1 & l_2^1 & \dots & l_N^1 \\ l_1^2 & l_2^2 & \dots & l_N^2 \\ \vdots & \vdots & \dots & \vdots \\ l_1^k & l_2^k & \dots & l_N^k \end{bmatrix} \tag{3.2}$$

where each column of the matrix corresponds to the load state at the particular time-snap, and each row of the matrix corresponds to the time series load value of the specific individual load.

According to the real situation of each sub-region, we can define a threshold $\tau_s \in [0, 1)$ to select the eliminated dataset A_{Fi} of individual loads that need to be considered in each sub-region:

$$A_{Fi} = \{\mathbf{x}_j \in A_i | \bar{\mathbf{x}}_j \geq \frac{\tau_s}{k} \sum_{i=1}^{k} \bar{\mathbf{x}}_i, j \in 1, 2, \dots k\} \tag{3.3}$$

Then, the individual loads of the eliminated dataset A_{Fi} can be renumbered from 1 to m:

$$A_{Fi} = \begin{bmatrix} \mathbf{x}_1(t) \\ \mathbf{x}_2(t) \\ \vdots \\ \mathbf{x}_m(t) \end{bmatrix}, m <= k \tag{3.4}$$

Centralization:

In the matrix A_{Fi} each row $\mathbf{x}_i(t), i \in 1, 2, \dots, m$ corresponds to a time series of one individual load, which includes two statistical information: mean and variance. The next step of our methodology, dimension reduction (due to functional principal component analysis (FPCA), seeing next subsection) [61], highly depends on variance and does not require information about the mean. Therefore we need to centralize each row of the eliminated dataset A_{Fi} to obtain each row having zero mean. The centralized row $\mathbf{x}_i^{'}(t)$ is given by:

$$\mathbf{x}_i^{'}(t) = \mathbf{x}_i(t) - \bar{\mathbf{x}}_i(t), i \in 1, 2, \ldots, m \tag{3.5}$$

After the data preprocessing, an eliminated centralized dataset is obtained by a $m \times N$ matrix, noted as $\boldsymbol{X}_{m\times N}$:

$$\boldsymbol{X}_{m\times N} = \begin{bmatrix} \mathbf{x}_1^{'}(t) \\ \mathbf{x}_2^{'}(t) \\ \vdots \\ \mathbf{x}_m^{'}(t) \end{bmatrix} \tag{3.6}$$

3.2.2 Preparing for Feature Extraction

The preparing for feature extraction is divided into two steps: dimension reduction and feature dissociation.

Dimension Reduction:

Instead of using the eliminated centralized dataset $\boldsymbol{X}_{m\times N}$, we conduct the dimension reduction of the data before the feature extraction. It can bring some benefits by doing this. Firstly, we can study the correlation among different components of the dataset and reveal the dataset's primary structure. It dramatically facilitates feature extraction and clustering. Secondly, during dimension reduction, it would usually reduce the noise and the redundant information in the dataset. Finally, it would significantly reduce the computational complexity for the classification process with the dimensionally reduced dataset.

Principal component analysis (PCA) [62] is one of the most widely used data dimensionality reduction algorithms, and the functional principal component analysis (FPCA) is an extended version of PCA [61]. The main idea of FPCA is to map the mean-centered individual load time series dataset from a higher m-dimensional state to a lower r-dimensional linear combination with orthonormal principal component harmonics by maximizing the variance within each principal component. The value of the resultant dimension r is selected according to the variance ratio of each principal component harmonics (also named eigenfunctions). Let the variance ratio of the dimension with the i-th most significant percentage be v_i, then

$$r = \arg\max_{r} \sum_{i=1}^{r} v_i \geq \tau_v, \tag{3.7}$$

in which τ_v is the variance ratio threshold. The value of τ_v is a hyper-parameter of our algorithm and is closely related to the dataset itself.

Feature Dissociation:

After dimension reduction by FPCA, we will obtain a weight matrix $\boldsymbol{W}$ and an eigenfunction matrix $\boldsymbol{E}$ to express the original dataset $\boldsymbol{X}_{m\times N}$ as:

$$\boldsymbol{X}_{m\times N} = \begin{bmatrix} \mathbf{x}_1^{'}(t) \\ \mathbf{x}_2^{'}(t) \\ \vdots \\ \mathbf{x}_m^{'}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} w_1^1 & w_2^1 & \dots & w_r^1 \\ w_1^2 & w_2^2 & \dots & w_r^2 \\ \vdots & \vdots & \dots & \vdots \\ w_1^m & w_2^m & \dots & w_r^m \end{bmatrix}}_{\boldsymbol{W}_{m\times r}} \cdot \underbrace{\begin{bmatrix} e_1^1 & e_2^1 & \dots & e_N^1 \\ e_1^2 & e_2^2 & \dots & e_N^2 \\ \vdots & \vdots & \dots & \vdots \\ e_1^r & e_2^r & \dots & e_N^r \end{bmatrix}}_{\boldsymbol{E}_{r\times N}} \tag{3.8}$$

where each row in $\boldsymbol{E}$ is one eigenfunction of the dataset $\boldsymbol{X}_{m\times N}$ and all rows are orthogonal with each other. Also, each row in $\boldsymbol{E}$ is a time series load data with the observation window N. Therefore, the time series of individual loads in each row of the original dataset $\boldsymbol{X}_{m\times N}$ is expressed as a linear combination of all rows of the $\boldsymbol{E}$ matrix. And each row in $\boldsymbol{W}$ is the weight vector corresponding to the individual load.

Eigenfunction matrix $\boldsymbol{E}$ includes exclusive features of the dimensionally reduced dataset. However, these eigenfunctions are challenging to interpret in the observation window $1, 2, \dots, N$, because they often exhibit shapes similar to polynomials from 1-order to r-order (due to keeping orthogonality). Therefore, we need to reflect the features of all eigenfunctions more clearly in the dimension of the observation window $1, 2, \dots, N$, that is, feature dissociation.

Varimax rotation (also called Kaiser-Varimax rotation) [63] is the most common way to do this with two main benefits: 1) keeping all eigenfunctions orthogonal; 2) keeping total energy constant. It maximizes the sum of the variances of the squared correlations between eigenfunctions and observation window time-snaps. It usually results in high eigenfunction values for fewer time-snaps and low eigenfunction values for the rest. In other words, the result is that a small number of time-snaps with important eigenfunction values are highlighted, which makes it easier to interpret. It gives a $N \times N$ rotation matrix $\boldsymbol{R}_V$ by:

$$\boldsymbol{R}_V = \arg\max_{\boldsymbol{R}} \left(\frac{1}{N} \sum_{j=1}^{r} \sum_{i=1}^{N} (\boldsymbol{E}\boldsymbol{R})_{ij}^4 - \sum_{j=1}^{r} \left(\frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{E}\boldsymbol{R})_{ij}^2 \right)^2 \right). \tag{3.9}$$

where $\boldsymbol{E}$ is the eigenfunctions matrix that we got from Section 3.2, and r is the number of rows of $\boldsymbol{E}$, and N is the number of columns of $\boldsymbol{E}$. Then, a new rotated eigenfunctions matrix $\boldsymbol{E}_R$ is obtained by:

$$\boldsymbol{E}_R = \boldsymbol{E} \cdot \boldsymbol{R}_V = \begin{bmatrix} \boldsymbol{e}_1'(t) \\ \boldsymbol{e}_2'(t) \\ \vdots \\ \boldsymbol{e}_r'(t) \end{bmatrix}_{r \times N} \tag{3.10}$$

All rotated rows in the new eigenfunctions matrix $\boldsymbol{E}_R$ are still orthogonal with each other, but the rotated eigenfunctions' variances on observation window time-snaps are highlighted.

After the above steps, feature extraction has received great benefits. At this time, the rotated eigenfunctions of each row in the rotated $\boldsymbol{E}_R$ are no longer arranged in descending order representing the variance of the data set. In the following algorithm, the data set variance represented by the rotated eigenfunction has been ignored. Still, the projection size of the time series curves of the individual loads on these rotated eigenfunctions is paid special attention.

The rotated eigenfunctions matrix $\boldsymbol{E}_R$ usually has two types of rotated eigenfunctions because the characteristics of individual loads may change over time, as mentioned in Section 1.2. One class will have massive loadings at some periods but tiny loadings at other periods. It is also called the short-term period with colossal loading. Another type will have relative statable loadings throughout the entire period. It is also called the long-term period with small loading.

Based on these two types of rotated eigenfunctions, this paper generated two classification processors: 1) segmental dataset processor; 2) full dataset processor. For the first one, we mainly classify the individual loads that have non-conforming properties on specific periods, such as DER users. For the second one, we specifically categorize the individual loads that keep the non-conforming properties all the time, such as the traditional industrial load. They have different methods for feature extraction, clustering, and classification and will be introduced separately below.

3.2.3 The Segmental Dataset Processor

This subsection mainly describes the classification approach for individual loads that only exhibit non-conforming characteristics in a short period. And we will use three parts to describe the method of feature extraction, clustering, and classification.

Feature Extraction:

For individual loads that only show non-conforming characteristics in a short-term period, the feature extraction methods mainly aim at the former of the two different types of rotated eigenfunctions mentioned above. Compared to the mean function, it has an extensive loading at some short-term period of the observation window and has minimal loading for the rest.

Before selecting the corresponding eigenfunction, we need to know which periods in the dataset's observation window we should focus on. From the *Regularity* and *Majority* of the three properties mentioned at the beginning of this section, we know that in the current power system, the conforming individual loads account for the majority and have similar performance. Therefore, the occurrence of non-conforming loads will bring more variances to the dataset to reflect the primary information in the data set. However, one of the advantages of the rotated eigenfunctions is that they reflect the prior information or variance in the dataset (no longer the variance in a single dimension, but the variances at different time-snaps in the observation window). Therefore, we can get further perception by computing the variance function of all rotated eigenfunctions,

$$V(s,t) = \frac{1}{r}\sum_{i=1}^{r}\left[\mathcal{E}_i(t) - \bar{\mathcal{E}}(t)\right]\left[\mathcal{E}_i(t) - \bar{\mathcal{E}}(t)\right],\; t = 1, 2, \ldots, N. \tag{3.11}$$

which actually treat all rotated eigenfunctions $\left[\boldsymbol{e}_1^{'}(t), \boldsymbol{e}_2^{'}(t), \ldots, \boldsymbol{e}_r^{'}t\right]^T$ from time-snap 1 to N as one random process note as $\mathcal{E}(n)$ including r independent processes. By calculating this variance function, we can easily draw the curve corresponding to the time variables t. Then the peaks and valleys on the variance function can be intuitively seen. If there is a peak at $t = n,\; n \in 1, 2, \ldots, N$, that means at n-th time-snaps of the observation window, and all rotated eigenfunctions have a huge variance, which is highly possible resulted from non-conforming individual loads. Therefore, the time-snaps like this are the periods in the observation window of the dataset that we should focus on. Then we could select some time-snaps corresponding to the peaks from the diagonal based on the actual situation, or select all peaks as your concerned time-snaps. Assuming y time-snaps from observation window $1, 2, \ldots, N$ are selected.

For one subset A_{Fi}, assuming we have a mean function of all individual loads time series $\bar{\mathbf{x}}^{'}(t)$ is given by:

$$\bar{\mathbf{x}}^{'}(t) = \frac{1}{m}\sum_{i=1}^{m}\mathbf{x}_i^{'}(t) \tag{3.12}$$

Each row (rotated eigenfunction) of the new matrix $\boldsymbol{E}_R$ is also a time series data noted by $\boldsymbol{e}_i^{'}(t), i \in 1, 2, \ldots, r$. We can define a set of square error row vector by:

$$\boldsymbol{d}_i(t) = \left(\boldsymbol{e}_i^{'}(t) - \bar{\mathbf{x}}^{'}(t)\right)^2 = \begin{bmatrix} d_1^i & d_2^i & \ldots & d_N^i \end{bmatrix},\; i = 1, 2, \ldots, r. \tag{3.13}$$

where each square difference row vector is a time-series data corresponding to the observation window time-snaps.

Feature extraction will be completed by selecting the corresponding rotated eigenfunctions for y focused time-snaps, respectively. For each focused time-snaps, we can find at least one (the more, the better) rotated eigenfunctions $\boldsymbol{e}_s^{'}(t), s \in 1, \ldots, r$ whose corresponding square difference vector $\boldsymbol{d}_s(t)$ has a significant value at the time-snaps but has a small value for the rest.

Clustering:

After extracting the features, we obtained y focused time-snaps of interest, and for each focused time-snap, some rotated eigenfunctions $\boldsymbol{e}_s^{'}(t),\ s \in 1, \ldots, r$ were selected. Therefore, we can analyze the clustering and classification of individual loads for each focused time-snaps.

Assuming for the j-th ($1 \leq j \leq y$) focused time-snaps, c rotated eigenfunctions we selected $\boldsymbol{e}_{sj_q}^{'}(t),\ q = 1, \ldots, c$. Because all rotated eigenfunctions were orthogonal with each other, a c-dimension orthogonal space is built by them. Therefore, each individual load in the subset will be uniquely mapped to a position in the space with a c-dimension coordinate using their projection on each rotated eigenfunctions, which is given by:

$$\boldsymbol{P}_i^j = \begin{bmatrix} \boldsymbol{p}_i^{j_1} \\ \boldsymbol{p}_i^{j_2} \\ \vdots \\ \boldsymbol{p}_i^{j_c} \end{bmatrix}_{1\times c} = \begin{bmatrix} < \mathbf{x}_i^{'}(t), \boldsymbol{e}_{sj_1}^{'}(t) > \\ < \mathbf{x}_i^{'}(t), \boldsymbol{e}_{sj_2}^{'}(t) > \\ \vdots \\ < \mathbf{x}_i^{'}(t), \boldsymbol{e}_{sj_c}^{'}(t) > \end{bmatrix}_{1\times c}, \ i = 1, 2, \ldots, m.\ j = 1, 2, \ldots, y. \tag{3.14}$$

where $\boldsymbol{P}_i^j$ is the position of the i-th individual loads in the orthogonal space of j-th focused time-snap.

According to the properties: *Majority* and *Regularity*, all the conforming individual load's positions in the orthogonal space will cluster together, and the non-conforming individual loads' position will be the outliers that are far from the center of clustering (CoC).

Classification:

To classify the conforming and non-conforming individual loads, we need to calculate CoC's position and determine the threshold as the distance from the CoC to classify the outliers out.

This paper defined the position of CoC of j-th focused time-snap as the mean coordinate of all mapped individual loads in the j-th orthogonal spaces, noted as $\boldsymbol{P}^j_{CoC}$:

$$\boldsymbol{P}^j_{CoC} = \begin{bmatrix} \boldsymbol{p}^{j1}_{CoC} \\ \boldsymbol{p}^{j2}_{CoC} \\ \vdots \\ \boldsymbol{p}^{jc}_{CoC} \end{bmatrix}_{1\times c} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{P}^j_i \tag{3.15}$$

Considering the *proportion* of properties, based on $\boldsymbol{P}_{CoC}$, power system experts can give the size of the threshold τ^j_d according to actual conditions. So far, we can claim that i-th individual load is non-conforming at j-th time-snap if:

$$\|\boldsymbol{P}^j_i - \boldsymbol{P}^j_{CoC}\| > \tau^j_d,\ i \in 1, 2, \ldots, m.\ j \in 1, 2, \ldots, y. \tag{3.16}$$

3.2.4 The Full Dataset Processor

This subsection mainly describes the classification approach for individual loads that will maintain non-conforming characteristics for a long-term period (usually for the entire observation window). We will discuss the different feature extraction, clustering, and classification methods.

Feature Extraction:

In Section 3.2-B3 we mentioned that there are usually two rotated eigenfunctions in the new $\boldsymbol{E}_R$ matrix. Unlike the previous approach, this approach is mainly aimed at the latter, which has relatively stable loadings throughout the entire observation window.

This kind of rotated eigenfunctions exists because of the *Majority* and *Regularity* of the individual loads that are conforming for the entire observation window. Using this rotated eigenfunction can help us quantify the degree of conforming of individual loads within the overall observation window and then benefit the clustering process.

According to (3.13), we have a set of square error row vectors $\boldsymbol{d}_i(t),\ i \in 1, 2, \ldots, r$, where each vector is the square error between $\boldsymbol{e}_i^{'}(t)$ and mean function $\bar{\mathbf{x}}^{'}(t)$ given by (3.12). It is also time-series data corresponding to the observation window time-snaps.

The feature extraction will be completed by selecting rotated $\boldsymbol{e}_s^{'}(t),\ s \in 1, \ldots, r.$, which has the corresponding square error vector $\boldsymbol{d}_s(t)$. It has relatively stable error throughout all observation window time-snaps. In other words, there is no massive error among the entire observation window.

Clustering:

The selected rotated eigenfunction $\boldsymbol{e}_s^{'}(t)$ from the previous step can help quantify and define the degree of conforming characteristic of the individual time-varying loads within the entire observation window.

Therefore, the first step of clustering is to normalize all individual load time-series data to the unit energy and record the value of the scaling factor. For each row of the original dataset $\boldsymbol{X}_{m\times N}$, the normalization can be done by:

$$\mathbf{x}_{Ni}^{'}(t) = \frac{\mathbf{x}_i^{'}(t)}{\|\mathbf{x}_i^{'}(t)\|},\ i \in 1, 2, \ldots, m. \tag{3.17}$$

where $\mathbf{x}_{Ni}^{'}(t)$ is the normalized time-series function for i-th individual load. It is actually to change the l-2 norm of each row vector to one. After this, all normalized time-series functions will have energy equal to one.

Next, we can quantify the degree of conforming properties of individual loads by computing the summation of all projects between the normalized individual load vector and the selected rotated eigenfunction $\boldsymbol{e}_s^{'}(t)$. Assuming that we have selected totally c rotated eigenfunctions in the feature extraction step. Then, the quantified conforming characteristic $\mathcal{Q}_i$ for the i-th individual loads is given by:

$$\mathcal{Q}_i = \sum_{j=1}^{c} < \mathbf{x}_{Nj}^{'}(t), \boldsymbol{e}_{sj}^{'}(t) > . \tag{3.18}$$

Based on the quantified conforming characteristic for each individual load, we can sort them as a new rank from small to large:

$$\mathcal{Q}_1^* \leq \mathcal{Q}_2^* \leq \cdots \leq \mathcal{Q}_m^* \tag{3.19}$$

Classification:

The lower the quantified conforming value, the more non-conforming the individual load is based on the new sorted rank. A threshold needs to be determined to separate the sorted rank into two parts. Considering the *proportion* of properties, the size of the threshold τ_d can be given by power system experts according to actual conditions. So far, we can claim that the i-th individual load in the new rank is non-conforming for the entire observation window, if:

$$\mathcal{Q}_i^* \leq \tau_d,\ i \in 1, 2, \ldots, m. \tag{3.20}$$

3.3 Experiments and Results

In this section, we use some real data records of PG&E in WECC to verify our proposed algorithm with detailed information given in the following. Two types of datasets with different observation window sizes are used for different classification approaches.

For the segmental dataset processor that analyzes non-conforming properties of different periods, no labels exist to verify. Therefore, this paper presented a few examples for each class, and we provided some suggestions and comments to the industrial community regarding each class.

For the full dataset processor that analysis the entire performance of individual loads, we have a label list for individual loads of PG&E, which was subjectively generated by the experience of some experts. We used this label list to analyze the proposed approach's accuracy performance and demonstrate our approach's effectiveness by analyzing the impact of the hyper-parameter: threshold.

First of all, the basic information of loads of PG&E utility was introduced by Figure 3.3, which is the histogram of the ratio of individual loads to the entire aggregated load. There is a total of 1610 individual loads belonging to this utility. Figure 3.3 shows that a large number of individual loads have a very tiny mean value of load demand (individual load to aggregated load ratio smaller than 0.1%), which are too small to impact the operation of the control center. These tiny individual loads are usually treated as one package (summate together). Their aggregated load usually performs highly conforming properties, so there is no need to classify non-conforming load from them. However, a few large individual loads have a tremendous mean value of load demand. Only 2.98% of individual loads have accounted for 13.17% of the aggregated load. The data elimination of our approach is actually to keep these large individual loads. In this case study, 363 individual loads with a mean value of load demand larger than 0.1% are reserved.

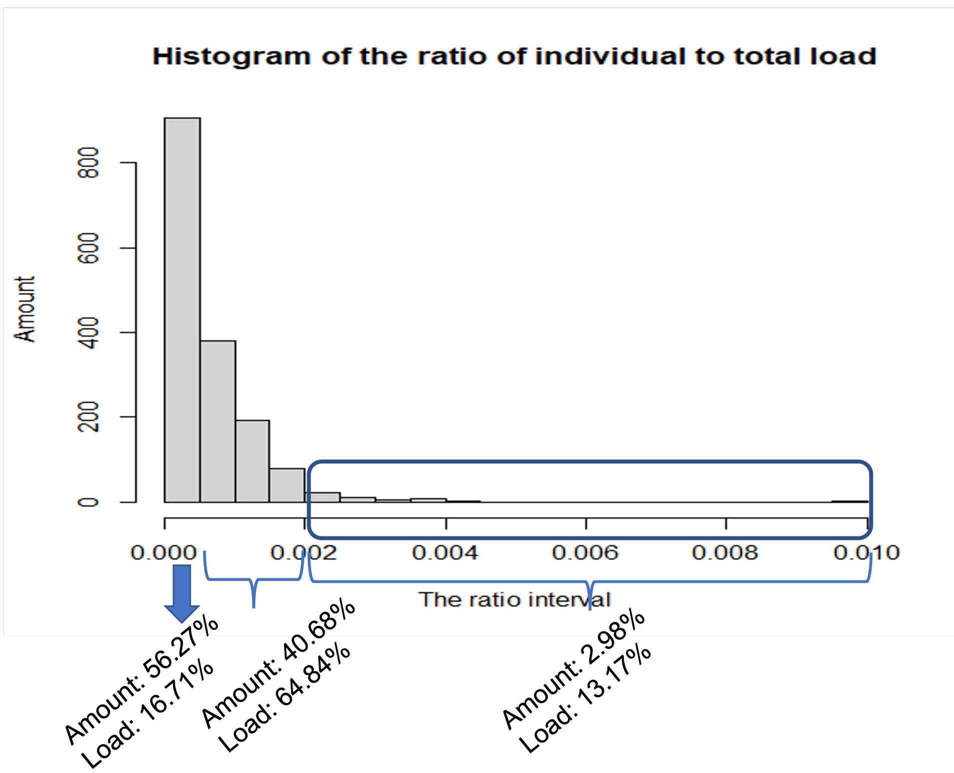

Figure 3.3: The Histogram of the ratio of individual loads to the aggregated load.

3.3.1 The Segmental Dataset Processor

Based on the eliminated individual load curve dataset, the next step of this approach should be the reduction of the dimension by implementing the FPCA method. This section uses a 5-mins interval load curve dataset for roughly 24 hours, including 363 eliminated individual loads as an example. To meet the conditions of using FPCA, the original dataset needs to be preprocessed, that is, to be centralized by subtracting the mean value from each individual load.

Each eliminated individual load has 279 timestamps in this dataset, so the original dataset size is 363×279. After implementing the FPCA method, the 279 eigenfunctions will be generated. Figure 3.4 shows the sorted results for the eigenfunctions by their covered variance of the original dataset. From the red line in the figure, we can see that the first 15 eigenfunctions have covered more than 98% of the variance in the original dataset. By retaining the first 15 eigenfunctions, we successfully reduced the original dataset in dimension.

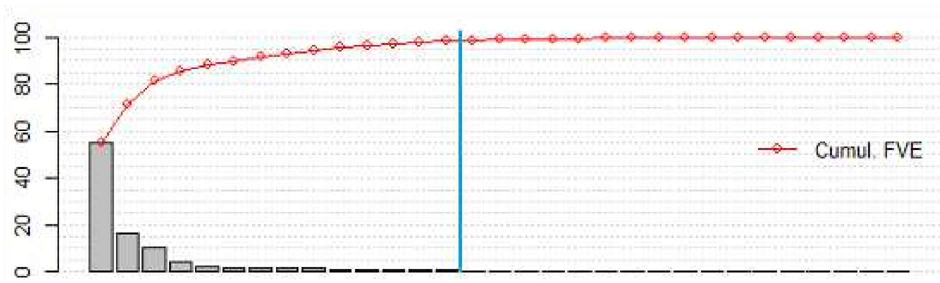

Figure 3.4: The variance of the sorted eigenfunction.

To understand the general information of the overall dataset, we can calculate the covariance surface by (3.11) using the remaining 15 eigenfunctions. s and t are different time stamps from 22:00 to the next day's 22:00. Figure 3.5 shows the results of the covariance surface, and the diagonal for $s = t$ means the variance of the overall dataset as a function of time. From the information expressed by the diagonal line in Figure 3.5, we can see that the differences among individuals of the entire dataset are mainly reflected in two time periods in this day: 2:00 - 4:00 and 14:00 - 16:00. Therefore, we consider analyzing these two time periods separately for daytime and nighttime.

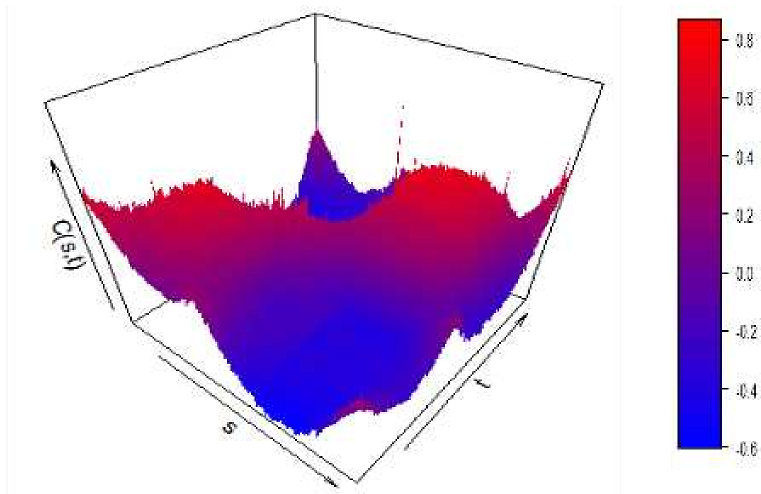

Figure 3.5: The covariance surface of the first 15 eigenfunction.

These 15 remaining eigenfunctions express the most significant variance in the original data set with a few dimensions as possible. The time dimension of each eigenfunction is often meaningless. The Varimax rotation algorithm needs to be implemented on the remaining eigenfunctions to focus on the interesting periods [63]. Its purpose is to maximize the variance of all eigenfunctions at each time point under the premise that the orthogonality of the original dataset is unchanged and the total energy summation is unchanged. It will result in a smaller function value on the period when the initial function value is relatively small and a more considerable function value on the period when the initial function value is rather significant.

The differences among individuals in the original dataset mainly concentrated in two periods, so it is easy to find a few new eigenfunctions that make substantial contributions to one of the two periods and weak contributions from the rotated eigenfunctions. So far, we further reduced the dimensionality of the original data set, and the new eigenfunctions have a different pertinence for different periods than the original eigenfunctions, thus completing the feature extraction.

To show the chosen rotated eigenfunctions, the rotated eigenfunctions are $+/-$ on the mean function of the whole individual load dataset. Figure 3.6 is showing a few examples for the chosen rotated eigenfunctions. The left three rotated eigenfunctions have a weak contribution in the night-time around 2:00 - 4:00 and have a comparatively substantial contribution in the daytime around 14:00 - 16:00. The right three rotated eigenfunctions have a weak contribution on the daytime

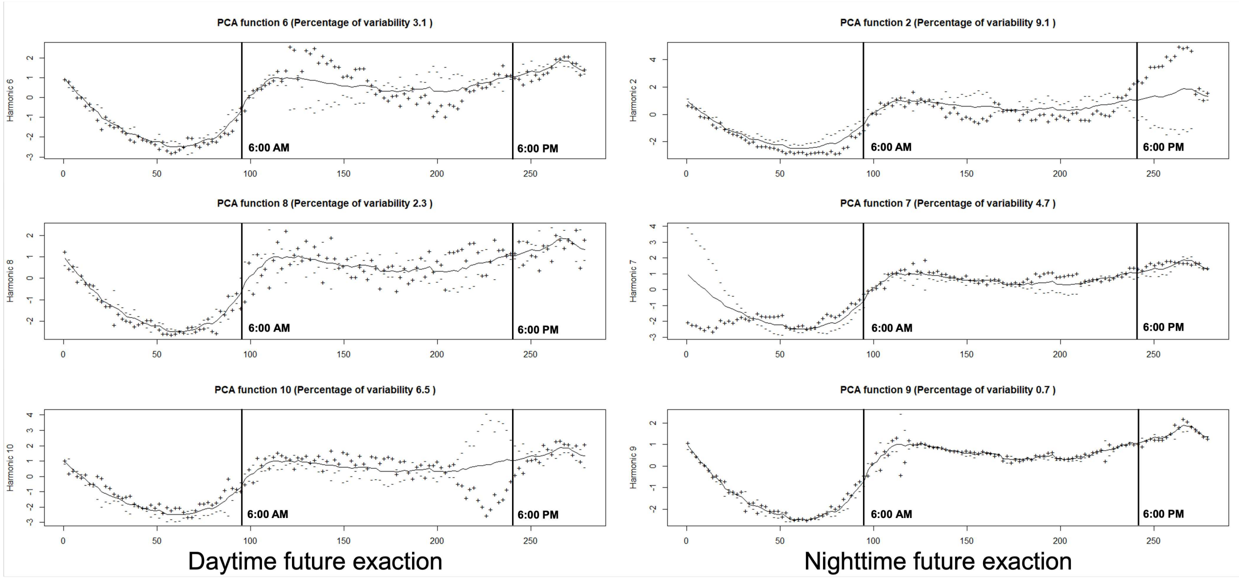

Figure 3.6: The examples for the rotated eigenfunctions for two different time periods.

around 14:00 - 16:00 and have a comparatively vital contribution on the nighttime around 2:00 - 4:00.

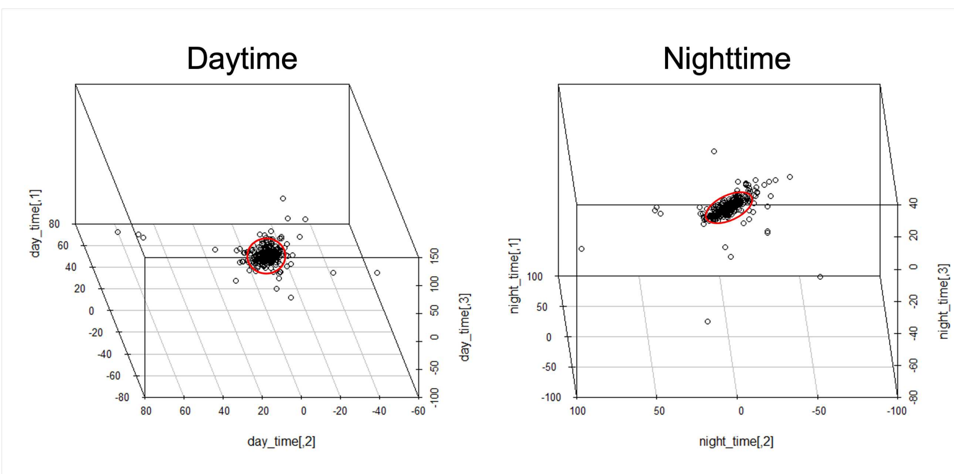

Figure 3.7: The clustering results for two different time periods.

Using the chosen rotated eigenfunctions that focus on different periods, the scores on each of the chosen rotated eigenfunctions can obtain the clustering results. To facilitate the reader's experience, this paper only selects three eigenfunctions for two time periods. They form two 3-D orthogonal spaces, respectively, to intuitively see the clustering results in Figure 3.7. In practical applications, using as many rotated eigenfunctions that meet the requirements simultaneously will improve the performance of clustering.

Based on the clustering results from above, this paper defined the center of clustering (CoC) as the mean coordinate of all mapped individual loads in the new orthogonal spaces. Then, we can define d as the Euclidean distance from the CoC coordinate in the orthogonal space, and all individual loads whose d is less than a threshold τ_d can be classified as a conforming load. However, the length of τ_d can only be determined by the experience of the expert in the power system.

Table 3.1: The classification results for two different time periods.

Nighttime conforming	Daytime confirming	Description
0	0	Either daytime and nighttime are non-conforming. For example, the traditional industrial loads.
0	1	Daytime conforming but nighttime non-conforming. For example, the night working residential loads that only has non-conforming property at nighttime.
1	0	Daytime non-conforming but nighttime conforming. For example, DER users, they will use energy from rooftop solar panel during daytime.
1	1	Either daytime and nighttime are conforming. For example, the traditional residential loads.

Figure 3.1 shows the classification results when treating the 24 hours data as two different periods: daytime and nighttime. Therefore, we classify all individual loads in PG&E into four categories: 1) either daytime and nighttime are non-conforming. For example, the traditional industrial loads. 2) daytime conforming but nighttime non-conforming. For example, the night working residential loads that only have non-conforming property at nighttime. 3) daytime non-conforming but nighttime conforming. For example, DER users will use energy from rooftop solar

panels during the daytime. 4) either daytime and nighttime are conforming—for example, the traditional residential loads.

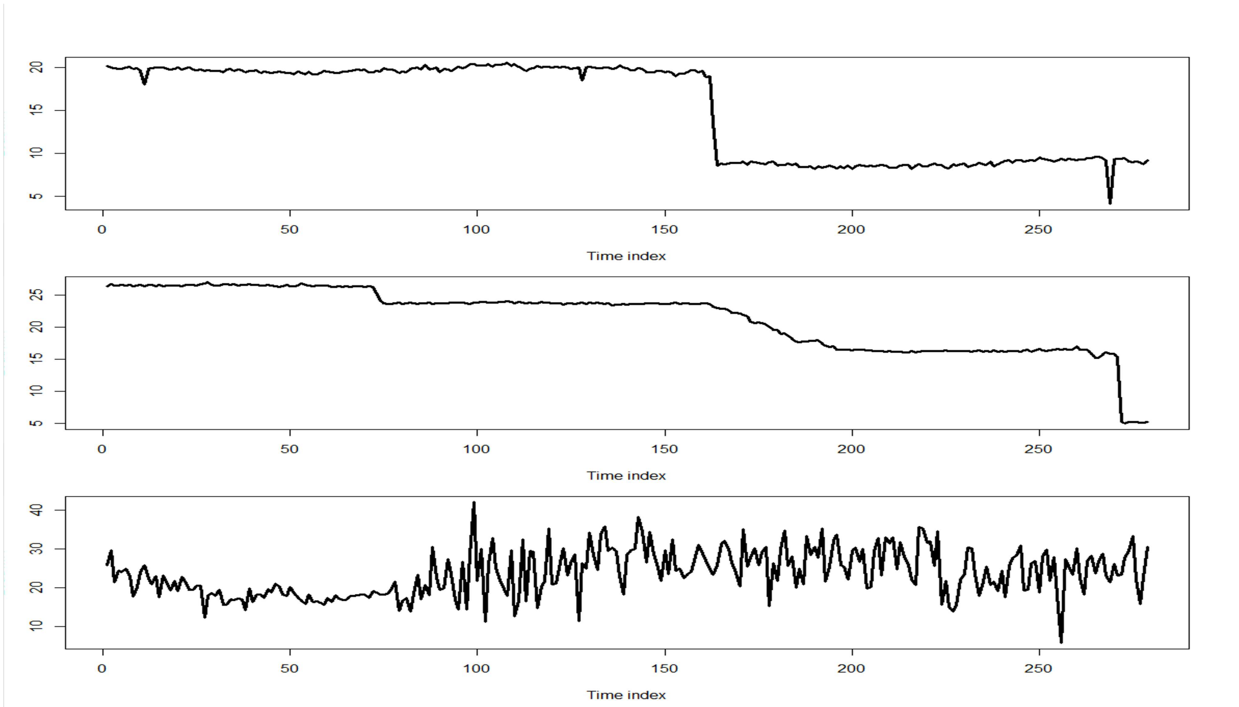

Figure 3.8: Examples for category (0,0), both nighttime and daytime are non-conforming.

To intuitively discern the difference between these four categories, From Figure 3.8 to Figure 3.11 presents three examples for each category. We can see from the figures that the load characteristics of the four categories are different.

3.3.2 The Full Dataset Processor

Based on the original individual loads' curve dataset, the first step to implement this approach should also be the reduction of the dimension by implementing the FPCA method. Therefore, we centralized the original dataset by subtracting its mean value from each individual load. This section uses a 5-mins interval load curve dataset for about one month, including 363 eliminated individual loads larger than 0.1% of the entire aggregated load. Each individual load consists of

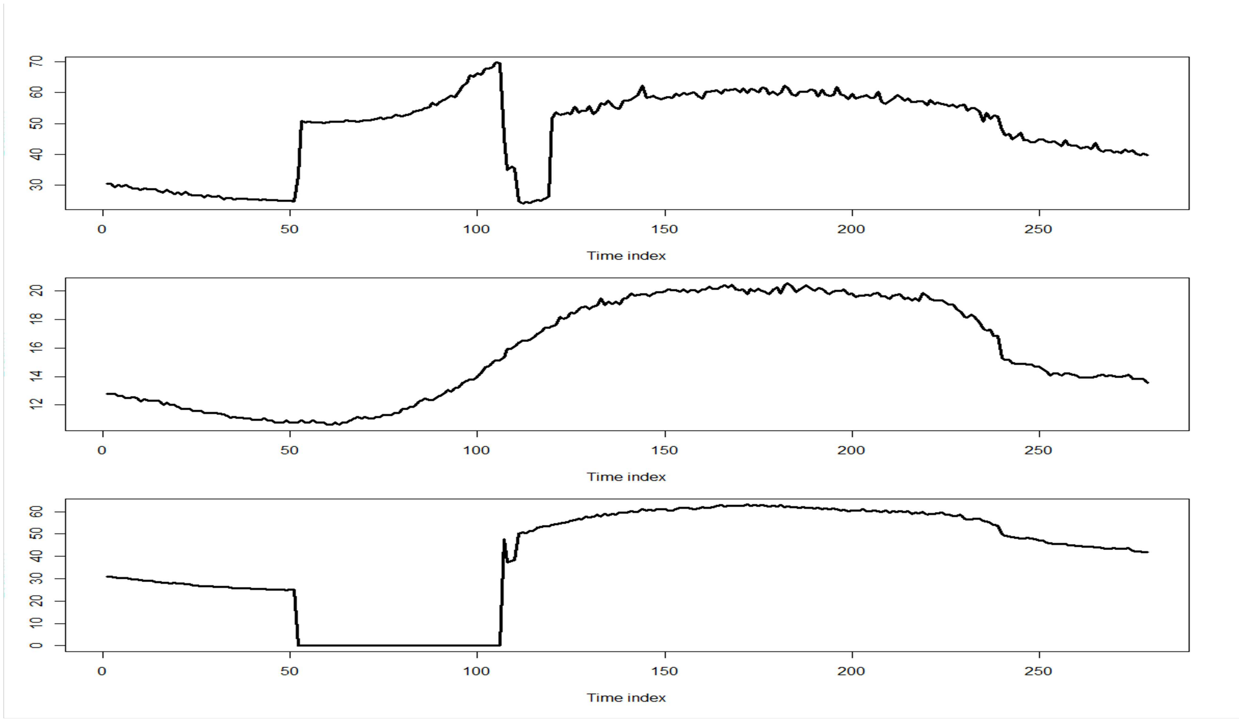

Figure 3.9: Examples for category (0,1), nighttime is non-conforming but daytime are conforming.

8640 timestamps. Therefore if we computed all the individual loads, the one-month data would require tremendous computation complexity beyond our current computation environment.

The original 8640 timestamps for each individual load were downsampled to 1750 timestamps using an anti-aliasing filter to improve the computational efficiency. Therefore, the size of the dataset should be 363×1750. After implementing the FPCA method, the 363 eigenfunctions will be generated. Figure 3.12 shows the sorted results for the eigenfunctions by their covered variance of the original dataset. From the red line in the figure, we can see that the first 32 eigenfunctions have covered more than 98% of the variance in the original dataset. By retaining the first 32 eigenfunctions, we successfully reduced the original dataset in dimension.

From the mean function shown by Figure 3.13 of all the individual loads, we can see the apparent seasonality, which is because of the regularity of the three properties that we mentioned at the beginning of this section. All conforming loads will maintain this characteristic, so this will become a powerful feature for us to classify conforming loads.

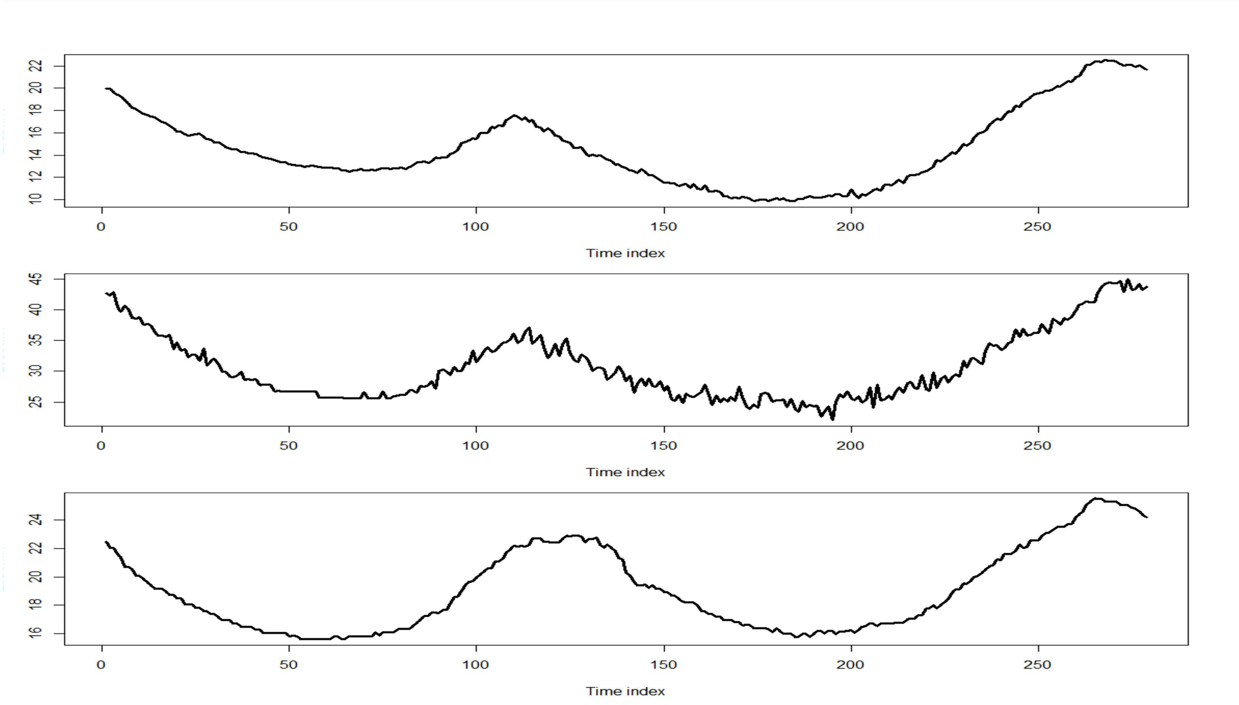

Figure 3.10: Examples for category (1,0), nighttime is conforming but daytime are non-conforming.

We also implemented the Varimax rotation algorithm on the remaining 32 eigenfunctions to extract the feature [63]. Unlike the previous search for the rotated eigenfunctions with significant contributions to different periods, we are looking for the rotated eigenfunctions whose contributions are stable during the entire period because this stable distributed eigenfunction represents the conforming characteristics of all individual loads. It can make those functions that contribute irregularly during the whole period more obvious. Figure 3.14 shows three examples for the chosen rotated eigenfunctions with a stable contribution for the entire period by $+/-$ each rotated eigenfunction on the mean function of all individual loads. In this way, we extracted the feature of the conforming load.

We can implement the clustering algorithm based on the extracted feature of conforming load by quantifying the conforming properties from the original individual load curve. Every individual load curve has a score regarding those chosen rotated eigenfunctions, and we defined the amount of conforming property as this score. All individual loads need to be normalized to the unit energy before quantifying the conforming properties. After summing the rotated eigenfunctions weighted

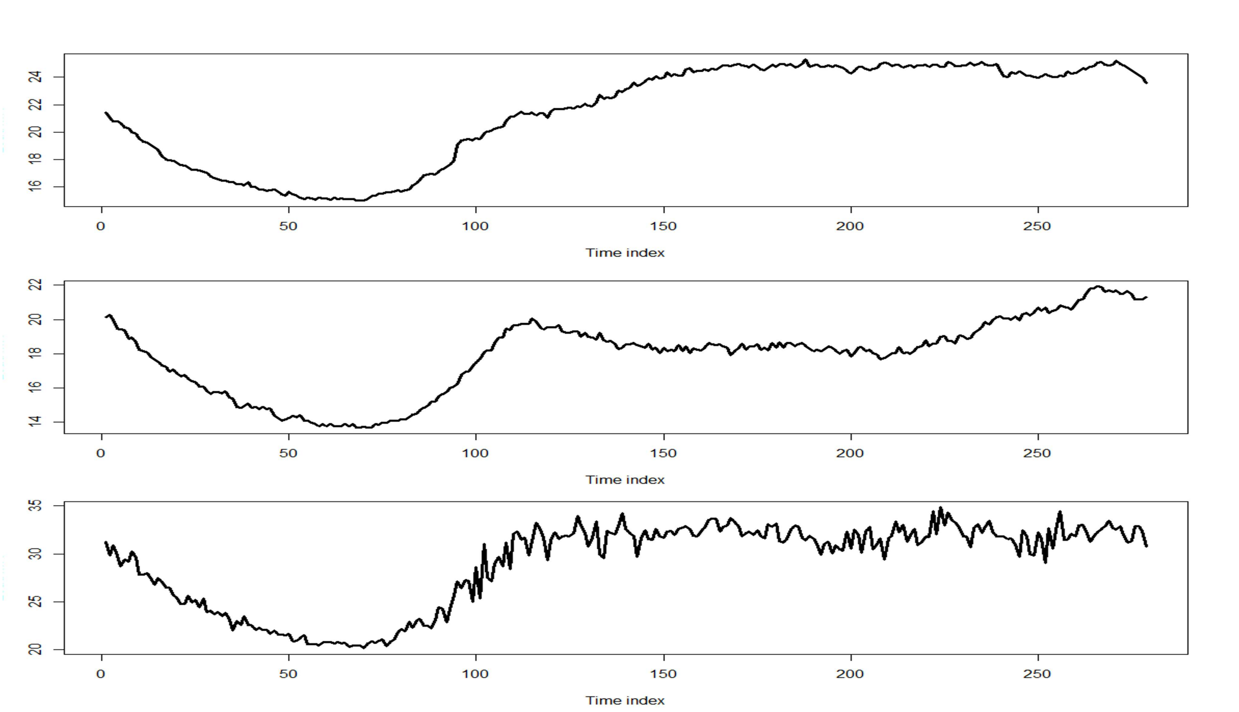

Figure 3.11: Examples for category (1,1), both nighttime and daytime are conforming.

by their scores from the unit energy individual load curve, we quantified the conforming component. The degree of non-conforming increases as the quantified conforming extent decreases.

Then, all individual loads can be sorted in one array by their conforming extent. A threshold value s in the range of this sequence could be defined to separate the sequence into two parts. The more significant part includes all conforming loads, and the minor part consists of all non-conforming loads. s is a hyper-parameter, which can be determined by the experience of the power system expert.

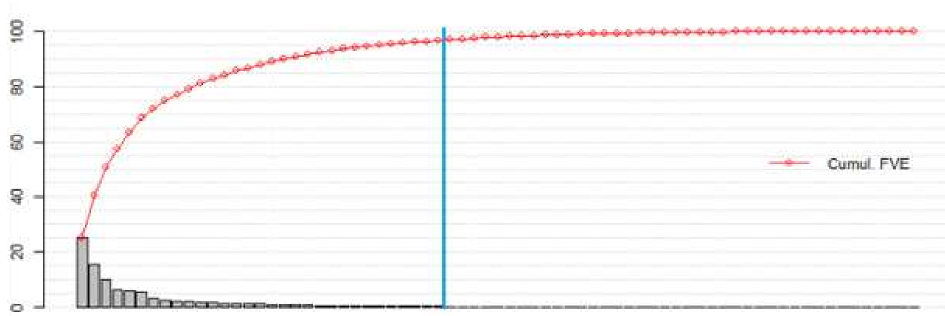

Figure 3.12: The variance of the sorted eigenfunction.

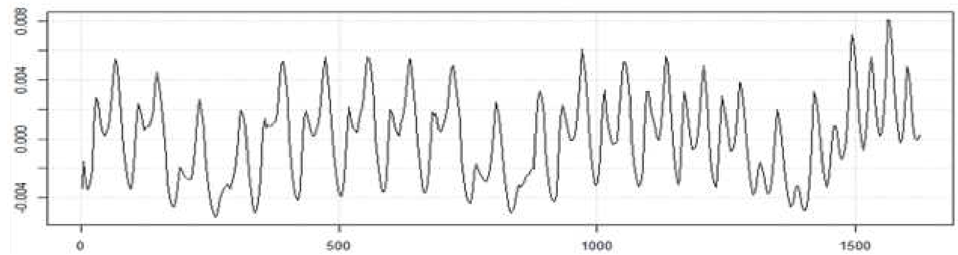

Figure 3.13: The mean function of all the individual loads.

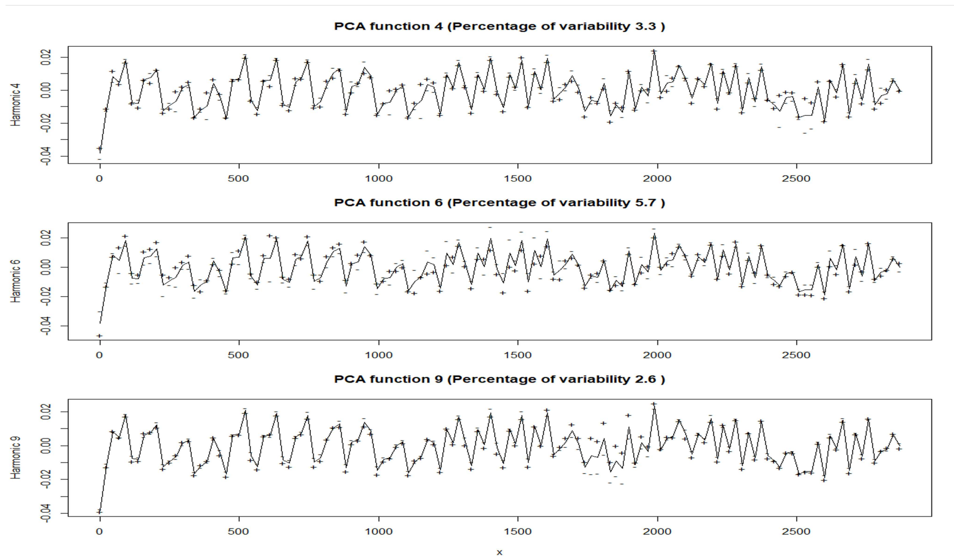

Figure 3.14: The examples for the rotated eigenfunctions for entire time periods.

To demonstrate the effectiveness of our method, we referred to a label list of all PG&E individual loads. This label list was subjectively generated by the experience of some experts in the control room, so it is not in complete confidence, also due to the vague definition of non-conforming load. This label list provided two unbalanced classes for 363 eliminated individual loads: 82 individual loads labeled as non-conforming and 281 ones labeled as conforming. Therefore, we introduced Matthews Correlation Coefficient (MCC) [64] to value the accuracy for this binary classification problem.

To analyze the impact of the hyper-parameter threshold, we use from the 1-st to the 363-th individual loads from the sorted sequence as the threshold to divide the sorted sequence into two categories and compute the MCC for each time. As Figure 3.15 shows, the highest performance occurs when using the 88-th individual load from the sorted sequence as the threshold. It means that our algorithm identified 88 individual loads as non-conforming and 82 labeled by the experience of experts. Furthermore, it shows that if we know the approximate proportion of non-conforming load in the entire data set, we will get a relatively reasonable classification result. Fortunately, this is true, and this paper summarizes it as one of the three properties-*proportion* when considering non-conforming loads problems.

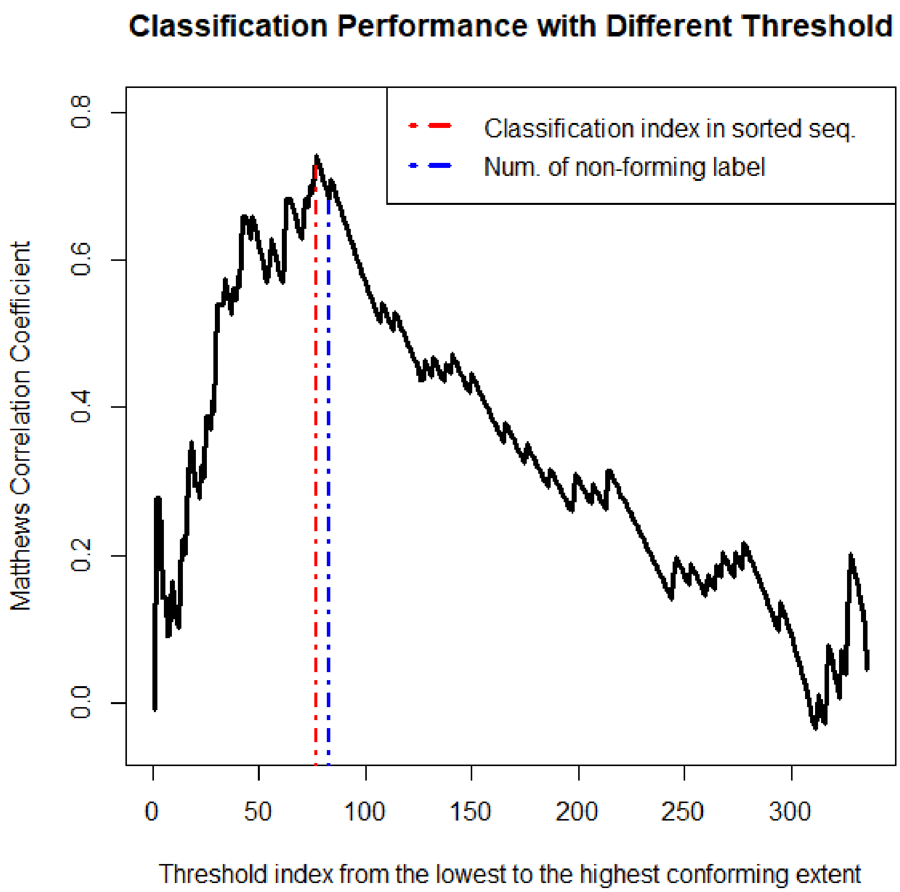

Figure 3.15: The classification performance with different threshold.

Figure 3.16 shows the classification results based on the overall performance of the individual loads from the entire period of one month. After implementing the full-time classification approach, all individual loads will be sorted by the qualities of the remaining non-conforming component in one sequence. The individual load with a smaller value in the order means fewer non-conforming characteristics and more conforming characteristics. In comparison, the meaning of the individual load with a more significant value in the order is the opposite.

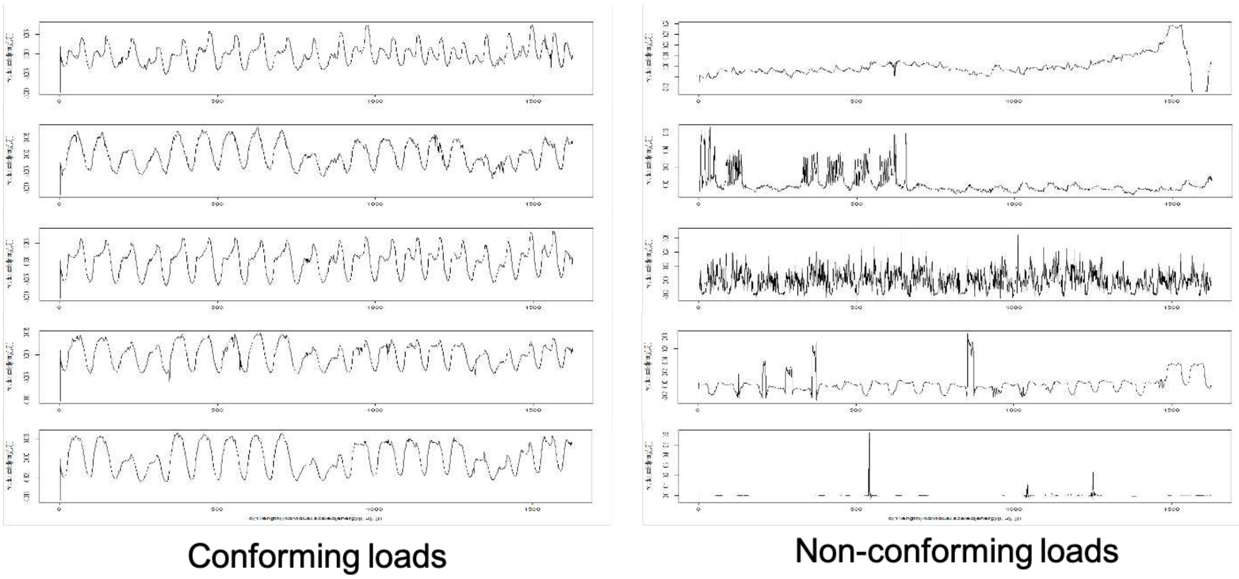

Figure 3.16: The examples for the classified conforming & non-conforming load for one month observation window.

To more intuitively reflect the clustering analysis results, this paper selects five individual loads with the minor non-conforming characteristics (the left column of Figure 3.16)) and five individual loads with the most significant non-conforming features (the right column of Figure 3.16) from the ranking. From the results, we can see that the individual loads in the left column maintain a relatively regular change characteristic within a month, while the individual loads in the right column reflect the rather irregular non-conformity characteristics, which demonstrates the effectiveness of our approach.

3.4 Summary

In this chapter, to solve the problem of non-conforming load classification, it first reduces the original data set using the FPCA method. Then considering the complex structure and vague nature of modern loads, this article proposes two classification processors: 1) the segmental dataset processor. 2) the full dataset processor.

The former can perform clustering analysis for the different characteristics of all individual loads in different periods. In displaying the results, we used the non-conforming characteristics of the two time periods to divide individual loads into four categories. We showed the differences between the different categories. Compared to the mean function, we could give some suggestions for these four categories. $(0, 0)$ means industrial loads that are keeping non-conforming property all the time. $(0, 1)$ implies some night-working residential loads that only have non-conforming property at nighttime. $(1, 0)$ means some DER users will use energy from the rooftop solar panels during the daytime, and therefore they will have a lower load measurement by the control center during the daytime. $(1, 1)$ means the traditional residential loads.

The latter extracts and quantifies the degree of the non-conforming component of each load based on all individual loads' performance over the entire period. We divided the final classification result into two parts from the sorted sequence through the experience of experts. The experts in WECC could adjust the threshold according to the extent of non-conforming characteristics of loads they would concern. Some identified non-conforming loads were confirmed by the experience of experts, and other identified non-conforming loads were verified by checking their time series. By referring to a label list subjectively generated by experts' experience in the control room, we analyze the impact of the hyper-parameter threshold on the classification performance and demonstrate the effectiveness of our approach. It shows that if we know the approximate proportion of non-conforming load in the entire data set, we will get a relatively reasonable classification result.

Chapter 4

Regional High-Resolution Medium-Term Load Forecasting

To generate the aggregated regional hourly medium-term electricity demand, we first introduced the region segmentation method. Then for each region, we decompose the aggregated regional hourly load into three parts and model them differently: base-load, seasonal-load, and residual term. Two case studies of the 2019 load curve forecasting of CAISO and BPA are presented at the end of this section to demonstrate the effectiveness of the proposed method. Eventually, we discussed the impact of COVID-19 on people's daily life by analyzing the 2020 load forecasting for CAISO and BPA.

4.1 Methodology

Climate change has a crucial influence on people's daily lives. The load demand always reflects the consumer's daily lifestyle, so climate change is one of the most critical factors influencing medium-term load forecasting. Therefore, it is necessary to divide the vast area into several sub-regions according to similar climate factors. Here we take Western Electricity Coordinating Council (WECC) as an example, according to the climate zones provided by [60] to divide the North American west power grid into 12 different regions to illustrate this methodology, as shown in Figure 3.2.

For each region, the aggregated regional hourly medium-term load curve can be expressed as a decomposition model:

$$y(t) = B(t) + S(t) + R(t) \tag{4.1}$$

where the $B(t)$ is called base-load that expresses the medium/long-term load variation trend. The $S(t)$ is called seasonal-load that represents the seasonal or periodic load variation of both long&short-term load curve information. The $R(t)$ expresses the residual between the output aggregated regional load curve $y(t)$ and the corresponding measurement.

Figure 4.1 shows the idea of the modeling process for load forecasting. As (4.1) mentioned, we decomposed the original load demand records into three parts: base-load, seasonal-load, and residual-term. For each component, we input different information and are trained in different ways. We used the daily average temperature of central cities and the weekly coding of an entire year as base-load input and used multi-layer perceptron (MLP) to train. We used the daily min-max temperature of central cities and weekday coding of each day as inputs for seasonal-load and used long-short term memory (LSTM) to train. We treated the residual between the label and the summation of base-load and seasonal-load as time-series and used autoregressive moving average (ARMA) to model it. Finally, we assembled the output of each part to get the final forecasted medium-term load curve.

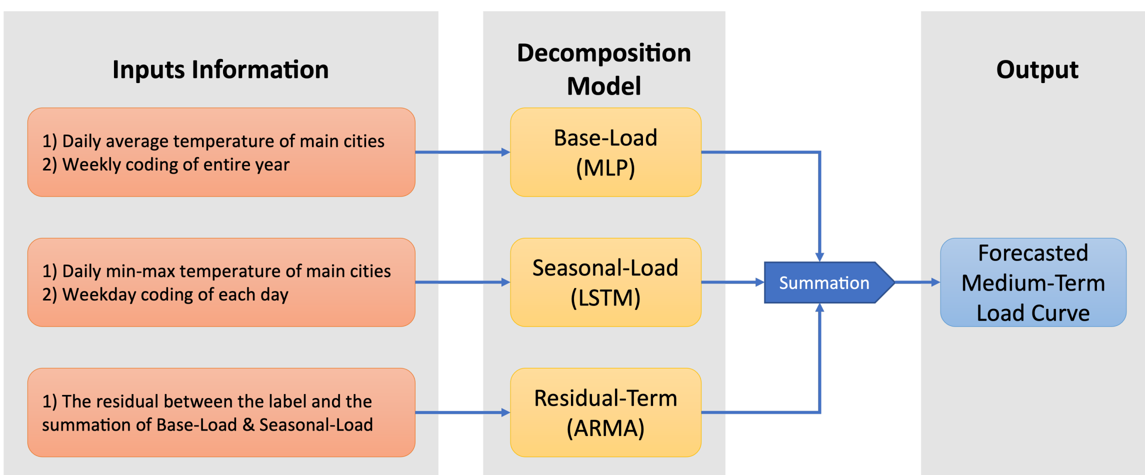

Figure 4.1: The idea of load forecasting model.

In the rest of this section, the details of modeling process of different parts are introduced bellow.

4.1.1 Base-Load Modeling

The modeling process of base-load consists of three steps: labeled data extraction, input matrix organization, and the learning method introducing.

Labeled Data Extraction:

Based on the data set introduced above, we should extract the measurement base-load curve from the original records to train the relationship between the input factors and the forecasting base-load. Because the $B(t)$ captures the medium/long-term load curve variation trend, it should keep the low-frequency component from the original yearly load curve.

Using Digital Fourier smoothing [65], we can accurately control how much low-frequency component needs to be kept. The cutoff point of the Fourier smoothing should be the number of weeks in the entire length of records. The low-frequency component is maintained based on the weekly repetition because the periodic characteristics of the load curve occur in ranging from daily, weekly to monthly [33]. We summarized the algorithm of Digital Fourier smoothing as below:

Algorithm 1: Digital Fourier smoothing algorithm

Data: The original aggregated load curve record.

Result: The extracted labeled base-load.

Initialization

Cutoff point: C = the number of weeks of the input load curve.

Length of input: L (set by user).

begin

Centralize the original load curve.

Add zeros if necessary to use the Fast Fourier Transform (FFT).

Implement FFT to the input curve.

if $C \leq i \leq L - C$ **then**

Set i-th data in amplitude of result of FFT $= 0$.

end

Implement inverser FFT.

Inverse centralize the load curve.

end

After extracting the low-frequency components from the original yearly load data curve, the labeled data for base-load is ready, noted as $B_l(t)$.

Input Factors Organization:

For medium-term load forecasting, some factors, such as population growth and economic change, have little influence and could be ignored, but the climate factor is significant. According to [33], the climate or weather conditions include temperature, humidity, wind speed, and cloud cover. Also, [33] claims that we can ultimately reduce all moisture, wind speed, and cloud cover to the temperature factor. Hence, we firstly consider the temperature factor as one of the input factors. Besides, the time factor always influences the load profiles because the time factor is directly related to people's daily lifestyles. Therefore, we eventually choose two factors as the inputs of the base-load $B(t)$: temperature and time.

The average temperature records of each day of main cities or populated areas are needed for temperature factors. Just average temperature is necessary because, as a trend-capturing part, base-load does not require the details of daily temperature change. However, we cannot use these average temperature data directly because the number of samples has not matched the length of the labeled base-load $B_l(t)$. The number of daily temperature samples will be smaller than the length of labeled base-load $B_l(t)$ because the sampling rate of $B_l(t)$ will remain extensive than one sample per day. Therefore, we must interpolate the original temperature sequence to the same length as the labeled base-load $B_l(t)$. To keep the smooth change and minimize the computation time cost, we use the Catmull-Rom splines interpolation to realize the up-sampling [66]. Assuming

M days, we collected average temperature, and the total length of the labeled base-load $B_l(t)$ is N. The algorithm can calculate the up-sampled temperature sequence:

Algorithm 2: The Catmull-Rom splines interpolation

Data: The temperature data for each day: $t(m), m = 0, 1, \ldots, K$

Result: The interpolated temperature series: $T(n)$

Initialization

Length of input temperature: $0, 1, \ldots, K$.

Required length of output: $0, 1, \ldots, N$.

begin

$\delta = \frac{K}{N}$, $s = 0$, $i = 2$, $m = 1$;

for n *from* 1 *to* N **do**

$s = s + \delta$;

if $s \geq i$ **then**

$s = s - i$;

$i = i + 1$;

$m = m + 1$;

end

$T(n) =$ Catmull-Rom(s, m);

end

end

Function *Catmull-Rom(s,m)* **is**

$$X(s) = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & \frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \\ 1 & -\frac{5}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} t_{m-1} \\ t_m \\ t_{m+1} \\ t_{m+2} \end{bmatrix}$$

return $X(s)$

end

After implementing Algorithm 2, the temperature sequence will have the same length as the labeled base-load $B_l(t)$. If there are multiple main cities or populated areas in this region, we could add numerous temperature records sequence.

For the time factor, to relate time change to the base-load curve change for each year, we introduced time coding to label 52 weeks of the year. Time should be fair to each sample, and the value of a time variable should not mean a certain extent. Thus, the input variable of the time factor should not be a rational number variable but a categorical variable. Also, because we consider keeping the low-frequency information in weekly repetition, all 52 weeks in a year were labeled by time coding, showing as Figure 4.2:

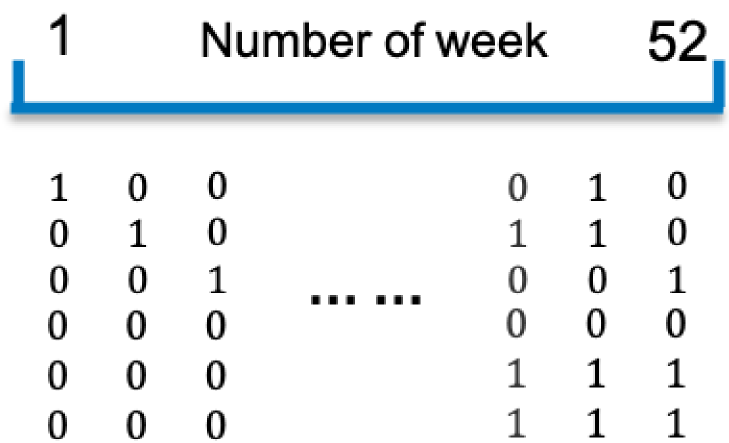

Figure 4.2: The weekly time coding.

Thus, we transformed a time variable into six parallel binary variables. To have the same length as the labeled base-load $B_l(t)$, we need to make all samples that belong to the same week from $B_l(t)$ have the same weekly time coding variable. Then, all input factors are collected and assembled with the same length as the labeled base-load $B_l(t)$.

Multi-Layer Perceptron (MLP) Learning Method:

After testing various neural network models, the multi-layer perceptron (MLP) [67] is the most suited learning method for base-load. The reason is that it keeps the nonlinear relationship between input factors and the labeled base-load $B_l(t)$ and has the least computation cost. The multiple layers and the nonlinear activation make MLP different from a linear perceptron, and it can identify data that is not linearly separable [68].

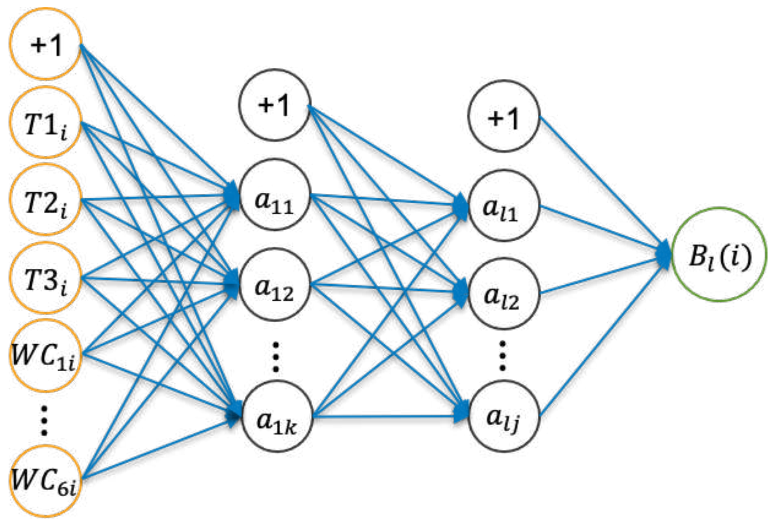

Figure 4.3: From input factors to the labeled base-load $B_l(t)$ by multi-layer perceptron.

In Figure 4.3, we assume that we considered the daily average temperature of three cities or populated areas, noted as $T1, T2, T3$, and the weekly time coding generate six binary variables, mentioned as $WC_1, WC_2, \ldots, WC_6$. Also, all input factors have the same length as the labeled base-load $B_l(i), i = 1, 2, \ldots, N$. From input-layer to output-layer, multiple hidden layers form a supervised learning technique called back-propagation for the training process [69].

4.1.2 Seasonal-Load Modeling

The modeling process of seasonal-load also has three steps: labeled data extraction, input matrix organization, and the learning method introducing.

Labeled Data Extraction:

To maintain the diversity of the original yearly load curve data set, we extract the labeled seasonal-load by subtracting the labeled base-load from the original records $L(t)$:

$$S_l(t) = L(t) - B_l(t) \tag{4.2}$$

After subtracting, the labeled seasonal-load $S_l(t)$ should centralize at 0, roughly symmetric fluctuating with it and changing amplitude.

Input Factors Organization:

For the same reason from the base-load above, the input factors of training seasonal-load are temperature and time, but there are some differences when building the input matrix for seasonal-load.

Unlike base-load, we introduced the temperature factors by the minimum and the maximum temperature records of each day of main cities or populated areas in the region. Because the seasonal-load $S_l(t)$ changes in peaks and troughs every day, the temperature changes in minimum and maximum could highlight the feature of seasonal changes better to match the training of $S_l(t)$. Thus, the resolution of temperature records for seasonal-load is two samples per day. We also need to implement Algorithm 2 (Catmall-Rom splines interpolation) to make the length of the temperature sequence the same as the length of seasonal-load $S_l(t)$.

Different form base-load, the time factors are introduced by weekday coding instead of weekly coding.

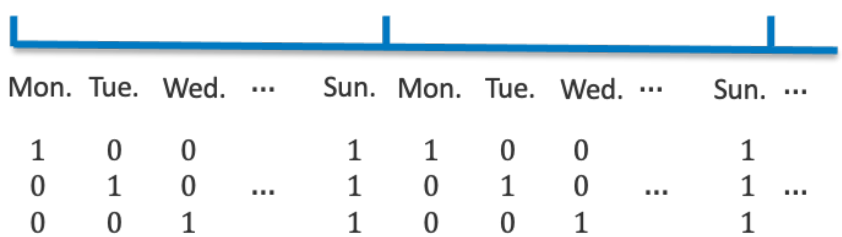

Figure 4.4: The weekday time coding.

As Figure 4.4 shows, the weekday coding cycle from Monday to Sunday will easily help the training process distinguish between working days and weekends. Also, as a rule of thumb, Thursdays tend to have the highest load peaks of one week, and this coding method could help the training process capture the periodic characteristic of seasonal-load for both long-term and short-term.

To have the same length as the labeled seasonal-load $S_l(t)$, we need to make all samples that belong to the same weekday from $S_l(t)$ have the same weekday time coding variable. Hence, for a similar reason of base-load, the time variable is transformed into three parallel binary variables. Then, all input factors are collected and assembled with the same length as the labeled seasonal-load $S_l(t)$.

Long Short-term memory (LSTM) learning method:

The labeled seasonal-load $S_l(t)$ is the curve centralized at 0 and roughly symmetric, fluctuating with it, and has a changing amplitude. The Long short-term memory (LSTM) is a kind of recurrent neural network (RNN) [70], which can relate the current output not only to the current input but also to the long- or short-term past inputs. LSTM can capture the periodic seasonal-load curve in the daily, weekly, and monthly load.

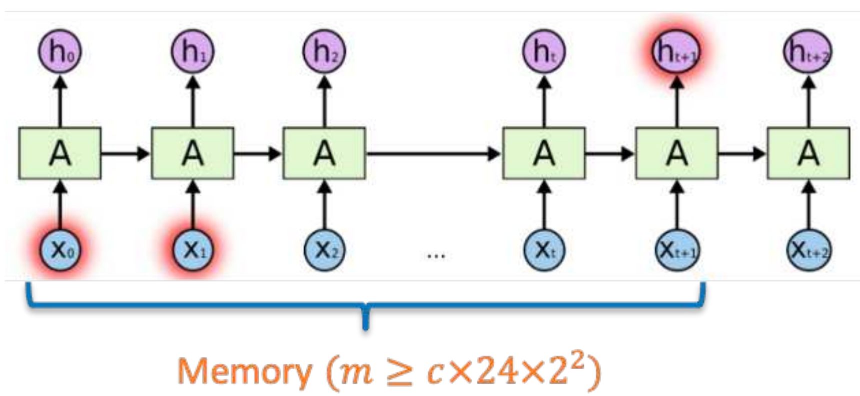

Figure 4.5: The idea of long short-term memory. [1] If the load curve sampling rate is c samples per hour, then the length of memory of $S_l(t)$ should larger than $c \times 24 \times 2^2$.

As Figure 4.5 shows, the current output (also called a hypothesis) is related to the current input and the past inputs. When using the LSTM, the input matrix has three dimensions: the number of input factors, training samples, and memory length. There are many ways for implementing the LSTM learning method, and we use it with a fixed size of memory. The number of input factors should be the number of prominent cities that collected temperatures plus their weekday coding binary variables. The number of training samples is equal to the length of the labeled seasonal-load $S_l(t)$. Each sample in the labeled seasonal-load sequence corresponds to a complete set of input factors. Each set of input factors includes a piece of memory of the corresponding data. We noted the length of memory as m. In our proposed method, the size of memory m should at least satisfy $c \times 24 \times 2^2$, where c is the sampling rate of the seasonal-load curve with unit samples per hour. The reason for this is to make the two adjacent weekdays have three different binary time variables. As Figure 4.6 shows, the input matrix should be this, assuming that the temperature for the three main cities is collected. The index of input samples s should be $m < s < n$, and each sample includes $s - m$ historical samples as the memory.

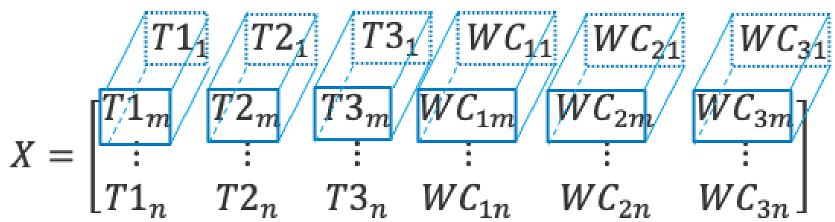

Figure 4.6: The idea of long-short term memory.

4.1.3 Residual-Term Modeling

The modeling process of residual-term has two steps: residual computation and ARMA modeling.

Residual computation:

After training the base-load and the seasonal-load by their model and labeled data, we obtained the base-load output and the seasonal-load output of the training process. Then we can get a preliminary assembled output by summing them together:

$$y'_{train}(t) = B_{train}(t) + S_{train}(t) \tag{4.3}$$

Then, we can compute the residual by the difference between the preliminary assembled output and the original load records:

$$R_{train}(t) = y_{train}(t) - y'_{train}(t) \tag{4.4}$$

ARMA modeling:

We treated the residual as a time-series. Before we model the residual by ARMA, we need to do some simple analysis to see if the residual is stationary enough. Usually, the residual generated by our method will centralize at zero and loss trend, and it usually has a little seasonality. Hence, the residual is usually robust stationary, and the ARMA model is used. Otherwise, if the residual is not stationary, we need to do a differencing operation for the raw residual to make it stationary, the ARIMA model.

There are many methods to check if a time-series is stationary or non-stationary. We summarized three standard methods [71]: 1) look at plots: one can review a time-series plot of the data and visually check any apparent trends or seasonality. 2) summary statistics: one can split the time-series into many partitions and compare the mean and variance of each group. If they differ and the difference is statistically significant, then the time series is likely non-stationary. 3) statistical tests: there are many tests, such as the augmented dickey-fuller test, that can provide a quick check and affirmative evidence that your time-series is stationary or non-stationary.

$$R_{train}(t) = c + \epsilon(t) + \sum_{i=1}^{p} \alpha_i R_{train}(t-i) + \sum_{i=1}^{q} \theta_i \epsilon(t-i) \tag{4.5}$$

(4.5) is the ARMA model where p is the order of the autoregressive polynomial, and q is the order of the moving average polynomial. α are the parameters of the autoregressive model, and θ are the parameters of the moving average model. ϵ are the error terms which is white noise. c is a constant.

Using the residual $R_{train}(t)$ that we computed from (4.4), we can solve c, α, θ for a specific order p, q of the ARMA equation.

4.2 Case Study

This section provided 2 case studies: the 2019 load curve forecasting and peak load prediction for California Independent System Operator (CAISO) and Bonneville Power Administration (BPA) utility. Eventually, by analyzing the load forecasting results for the 2020 year of CAISO and BPA, we discussed the impact of the COVID-19 on people's daily lifestyles. Figure 4.7 shows the distribution map of the Western Interconnection RCs. CAISO is referred to as CISO on the map. We can find the location of utilities on this map.

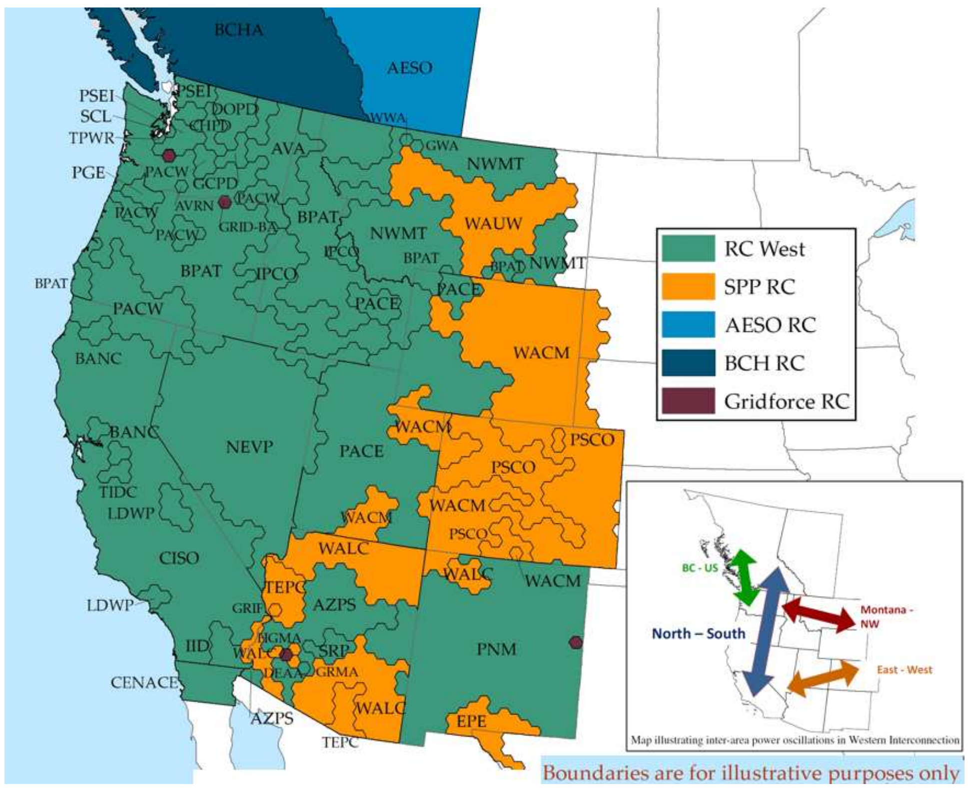

Figure 4.7: Western Interconnection RCs Footprints [Source: WECC].

4.2.1 Case study A: CAISO 2019

In this section, a case study of the hourly aggregated load curve for California Independent System Operator (CAISO) in 2019 is forecasted. We chose CAISO as an example because its entire region is the same climate zone and has the same independent power jurisdiction area. That means the data of the aggregated load for the whole region is easy to get. The data of electricity demand for CAISO from 2009 to present could download at [72].

For CAISO, due to the increase in air-conditioning usage by the hot climate, summer is the highest period of electricity consumption throughout an entire year. The peak electricity consumption week of a whole year also appears in summer. Thus, the resilience of the electricity system for CAISO is highly challenged in summer and usually has some active or passive blackouts occur in recent years' summer. This section used the aggregated load data records for CAISO from 2014 to 2018 to train our proposed model. Then the trained model was used for forecasting or generating the hourly aggregated load curve for CAISO August 2019 and forecasting the date that the peak load in 2019 occurs.

Also, we collected the input information of temperature from three main cities of California: San Francisco (SF.), Los Angeles (LA.) and San Diego (SD.).

Base-load Forecasting

The input matrix is first created. As the Section 4.1 mentioned that the daily average temperature records for SF., LA., and SD. are needed, and one can download them [73]. Figure 4.8 shows the 2019 daily average temperature curves for three main cities as an example, which is generated by implementing the interpolation algorithm introduced by Algorithm 1. After normalizing them and combining the weekly coding mentioned in Section 4.1.1, we created the input matrix.

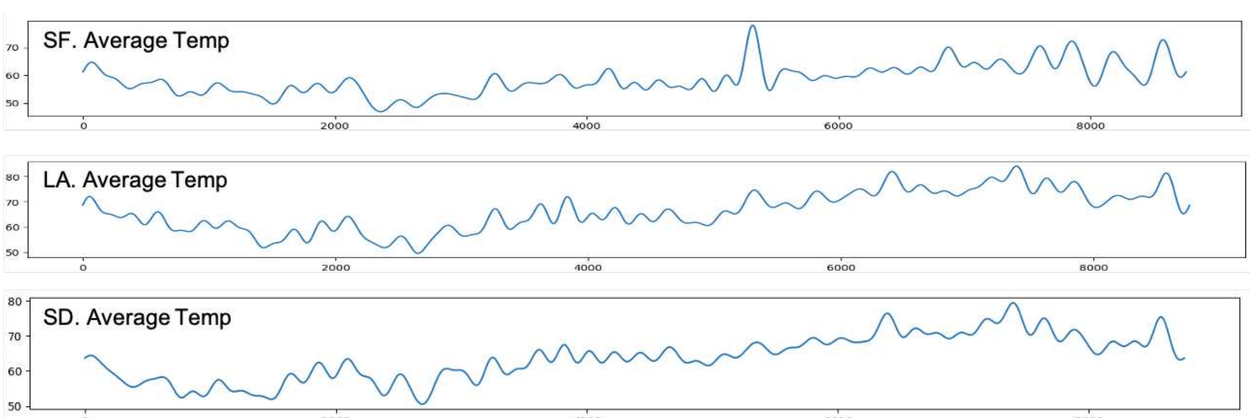

Figure 4.8: The interpolated 2019 average temperature curves of 3 main cities for base-load.

The training data set of base-load needs to be extracted for the next step. As Figure 4.9 shows, the time domain yearly load curve was transferred to the frequency domain by the digital Fourier transform (DFT), and we plotted the amplitude responses. In this amplitude response, there are some prominent peaks, and they intuitively explain the periodic characteristics of the load curve. These periodic characteristics of the load curves range from the daily load, the weekly, monthly, seasonal to yearly load curves [33]. Because we computed the DFT-transferred load curve on each year, the minimal high-amplitude cycle is one, which means one year and map to the highest peak in the middle of Figure 4.9. Moreover, we could discover that there will be a high amplitude for about every $\pm 52 (year^{-1})$. It is because each year has 52 weeks, and people's daily lifestyle highly repeats every week. Hence, we consider keeping the low-frequency information in weekly repetition by keeping the truncated at least $\pm 52 (year^{-1})$ in the frequency domain. In practice, people usually choose the absolute value of the truncated cycle a little larger than $52 (year^{-1})$ to have more tolerance, and in our research, $60 (year^{-1})$ is used (marked between two red lines).

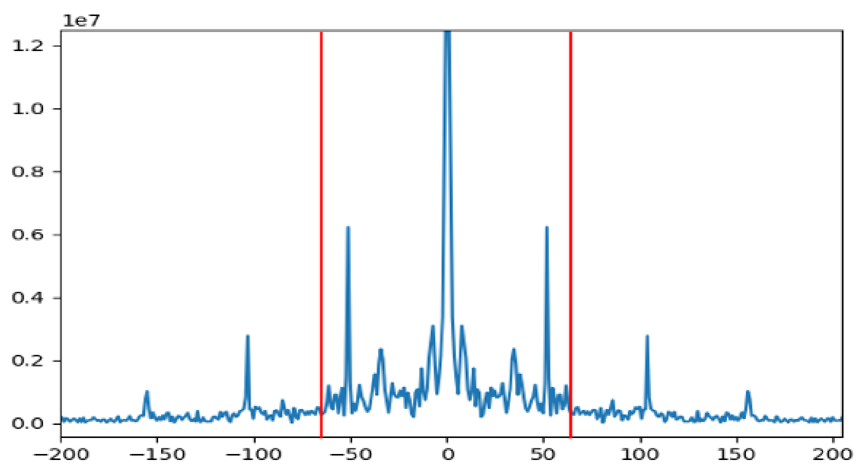

Figure 4.9: Digital Fourier transform of yearly load curve and extract the low frequency component.

By implementing Algorithm 1 with keeping frequency components lower than 60, we extracted the base-load. Figure 4.10 shows an example using CAISO 2019 load records. The blue curve is the original record in this figure, and the orange curve is the extracted labeled Base-load $B_l(t)$. People can observe that the base-load indicated the trend of the original records.

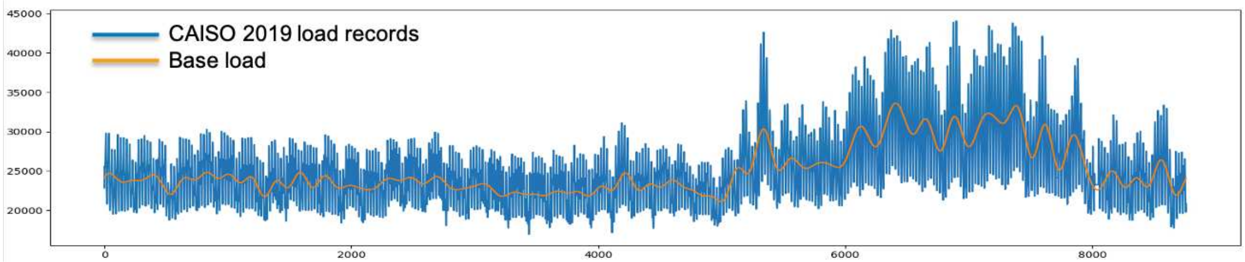

Figure 4.10: The base-load example for 2019 CAISO load records.

Using the same method to extract the labeled base-load from 2014 to 2018 original load data records, generates the labeled base-load data of the entire training set, shown as the orange curve in Figure 4.11. Then, using the same method to extract the labeled base-load from 2019 summer (from Jun. to Sep.) original load data records, generates the labeled base-load data of the entire test set, shown as the orange curve in Figure 4.12.

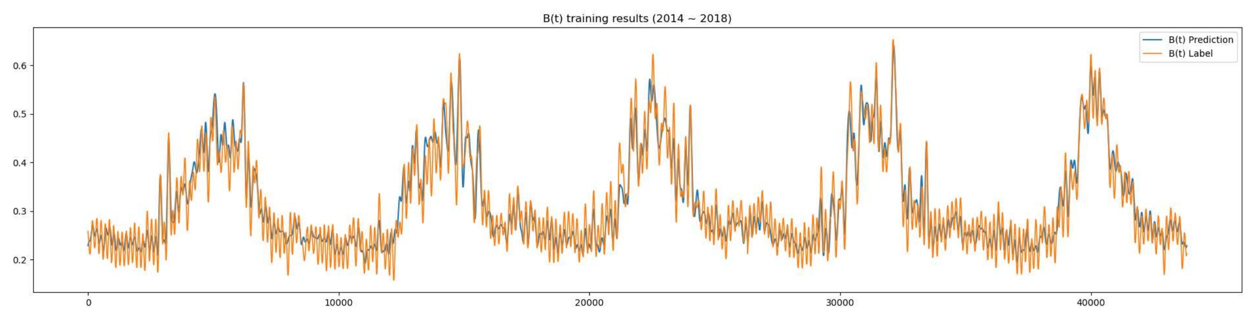

Figure 4.11: The training results for base-load.

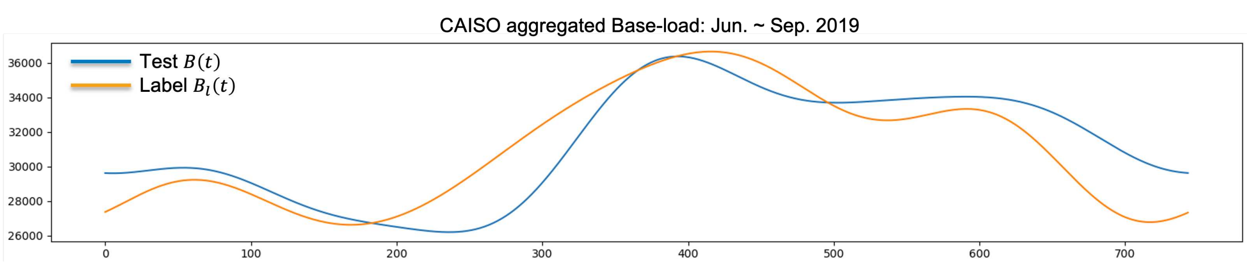

Figure 4.12: The testing results for base-load.

After learning with the MLP for the input matrix and the labeled base-load $B_l(t)$, we obtained the base-load $B(t)$ model. Using this $B(t)$ model, we generated the forecasting curve for the training set, shown as the blue curve of Figure 4.11, and we also developed the forecasting curve for the test set, shown as the blue curve of Figure 4.12. We will finally assemble it with the forecasting curve of seasonal-load $S(t)$ and the random error to generate the forecasting curve of the final output.

Seasonal-Load Forecasting

As the Section 4.1.2 mentioned the minimum and maximum temperature data for each day for SF., LA., and SD. are needed to create the input matrix for seasonal-load training. After implementing Algorithm 2, we will interpolate the minimum and maximum temperature data to a min-max temperature curve with the same length as the corresponding nature time of the base-load curve. Figure 4.13 shows the normalized min-max temperature curve for three main cities of August 2019 as an example. After consisting together with the weekday coding mentioned in Section 4.1.2, we created the input matrix of the seasonal-load training process.

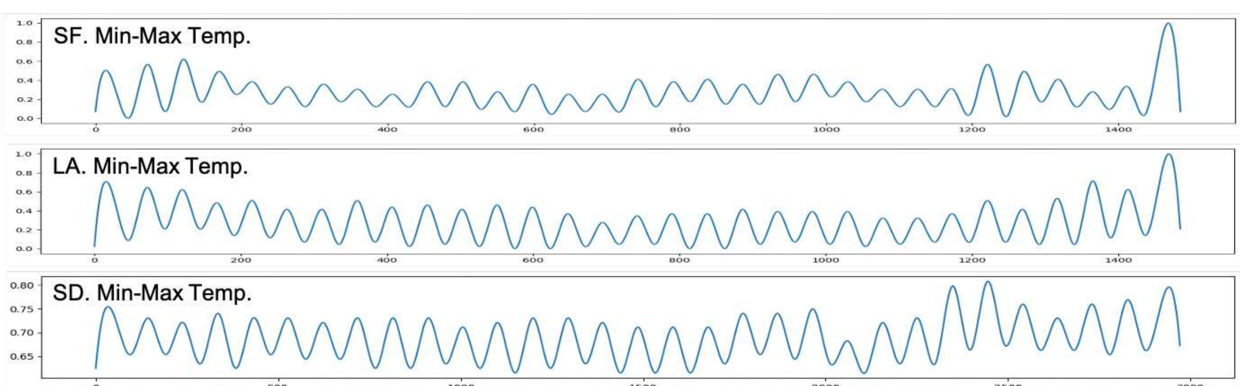

Figure 4.13: The interpolated 2019 Min-Max temperature curves of 3 main cities for seasonal-load.

We need to extract the training data set of seasonal-load for the next step. As Section 4.1.2 mentioned, the seasonal-load is extracted by subtracting the base-load from the original load records. Figure 4.14 shows the data from Jun. to Sep. of CAISO 2019 as an example, and it shows that the orange curve of the top plot is the extracted base-load $B_l(t)$, and the blue curve of the top plot is the original load records. After extracting the base-load from the initial load records, we generated the bottom plot of Figure 4.14 as the extracted labeled seasonal-load $S_l(t)$. As Figure 4.14 shows, the labeled seasonal-load $S_l(t)$ is centralized at 0 and roughly symmetric, fluctuating with it and changing amplitude.

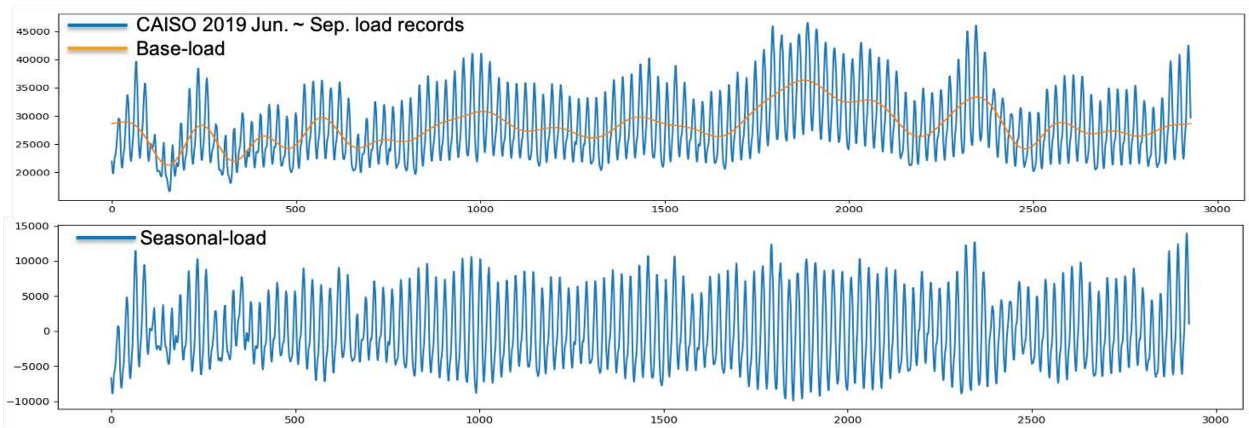

Figure 4.14: The seasonal-load example for 2019 summer CAISO load records.

The training and test results are presented below. Unlike the base-load, the seasonal-load focuses on the periodic changes based on each day, so the sampling rate of seasonal-load needs to be larger than the base-load to obtain higher accuracy. Because the goal of this case study is to forecast the summer load curve and the date of the peak load and the input information are temperature and the weekday, we could separate the August $S_l(t)$ from the $S_l(t)$ curves of 2017, 2018, and 2019. Then we spliced them together into a training sequence. This discontinuity caused by the splicing will disturb the LSTM learning method and make the accuracy of the curve lower at the date of splices. However, because the beginning and the ending of August do not influence the final goal of this case study, we used this splicing to violently reduce the amount of calculation based on the higher sampling rate. If it is necessary and possible, the continuous training sequence is better to use.

Figure 4.15 shows the learning results for the training process, and Figure 4.16 shows the learning results for the testing process.

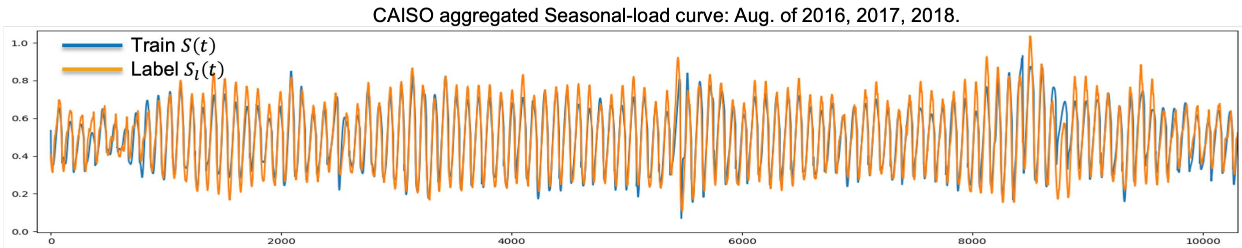

Figure 4.15: The training results for seasonal-load.

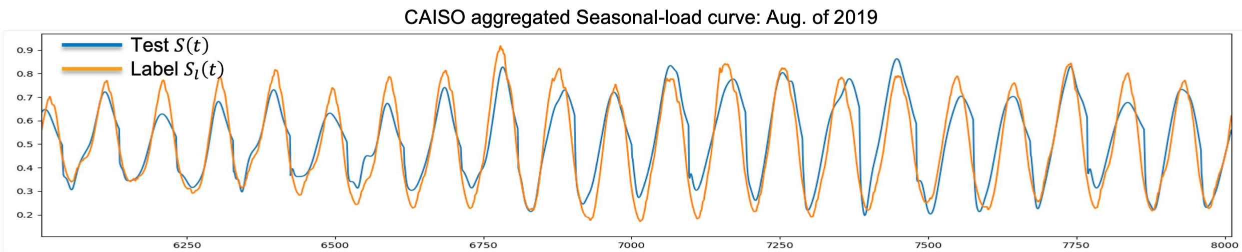

Figure 4.16: The testing results for seasonal-load.

Assembled Aggregated Load Forecasting

We can generate the final aggregated load forecasting by assembling them after obtaining the forecasting load curve for both base-load and seasonal-load. The forecasting results for CAISO August are plotted in Figure 4.17. In this figure, the orange curve is the recorded load curve, and the blue curve is the predicted load curve. As the figure shows, we successfully predicted the peak load week of 2020 for CAISO.

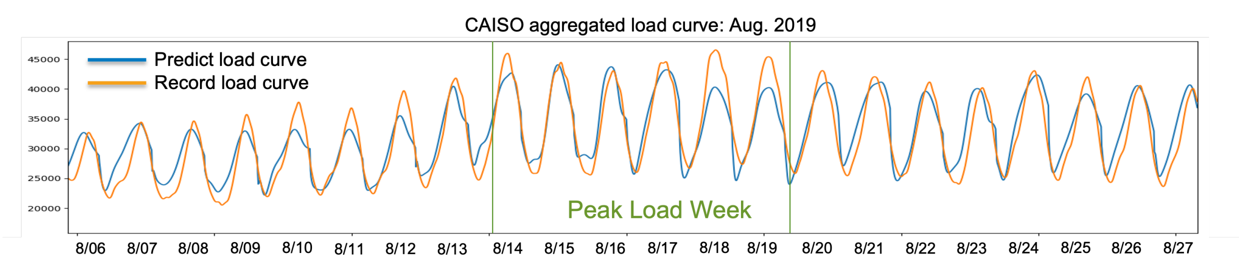

Figure 4.17: The aggregated load curve for Aug. 2019 CAISO.

Then, the predicted results for the peak load day are zoomed in, as Figure 4.18. From the figure, we can notice that the error of time of the predicted peak load occurs smaller than 1 hour.

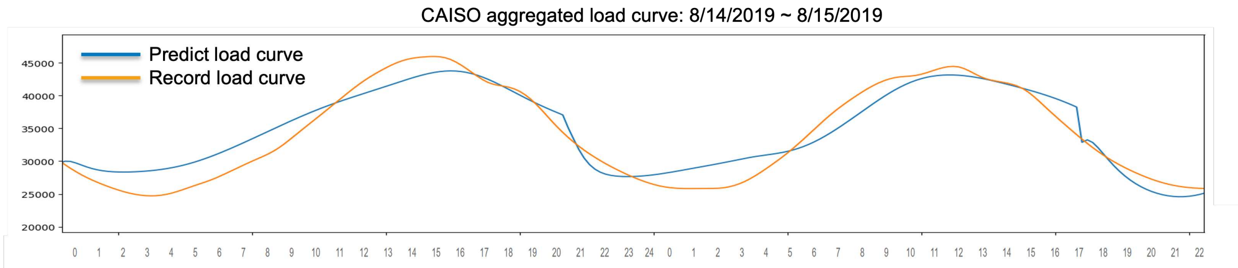

Figure 4.18: The aggregated load curve for the peak load date of CAISO.

4.2.2 Case study B: BPA 2019

This section used hourly load data for BPA from 2014 to 2018 to train our model and generate the 2019 load curve. The main reason for choosing BPA is that its entire region is the same climate zone and belongs to the same independent power jurisdiction area. Another reason is that the data for BPA is open source, which people can download at [74]. It means that the data of electricity data of the aggregated load is easy to get.

BPA area is north of the U.S.A, and its load profile is in sharp contrast with CAISO because the climate in the BPA area is freezing in winter, and people use lots of heaters to keep warm in winter. Therefore, the peak load always occurs during winter around January.

We collected input information on temperature from three main cities: Seattle, Portland, and Kennewick. In this section, we will present the results for base-load, seasonal-load, and aggregated output. Other details, such as temperature curve and frequency domain amplitude plot, will be omitted because these steps are similar to the previous CAISO case study.

Base-load Forecasting

The input matrix for training base-load is generated by combining the interpolated average temperature curve of Seattle, Portland, and Kennewick and the 6 bits weekly coding variables. The label data for base-load is also generated by Algorithm 1 to keep the low-frequency component.

Figure 4.19 and Figure 4.20 present the prediction results for the training set and testing set. From the figures, we can see that our approach provided good prediction results for BPA base-load. Also, we can see that the peak load of BPA always occurs in January.

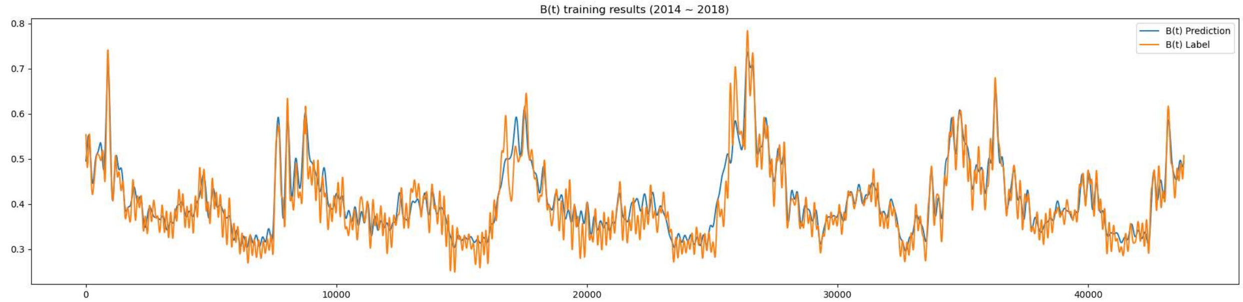

Figure 4.19: The normalized training results of base-load for BPA from 2014 to 2018.

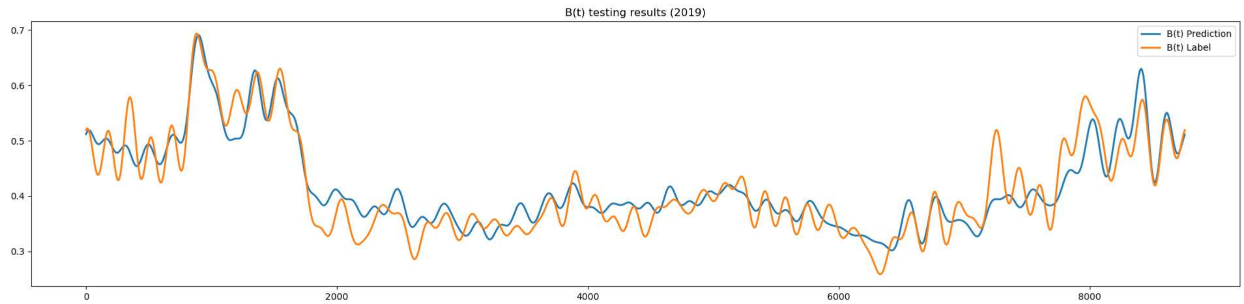

Figure 4.20: The normalized testing results of base-load for BPA 2019.

Seasonal-Load Forecasting

The input matrix for training seasonal-load is generated by combining the interpolated min-max temperature curve of Seattle, Portland, and Kennewick and the 3 bits weekday coding variables. We also developed the label data for seasonal-load by subtracting the base-load from the original records.

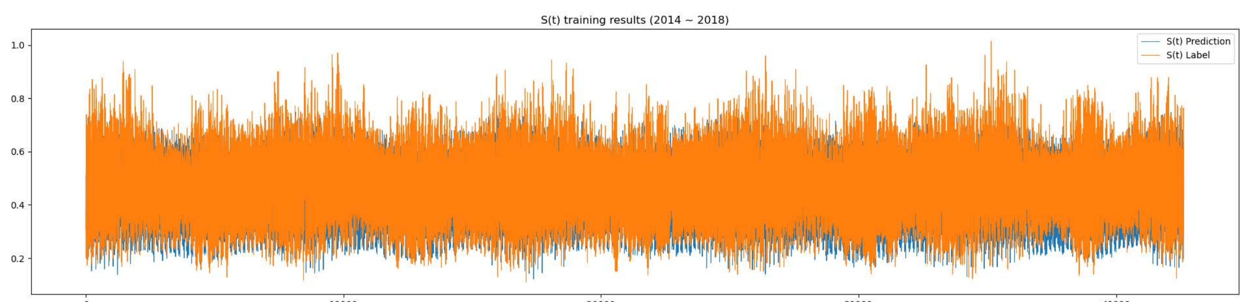

Figure 4.21: The normalized training results of seasonal-load for BPA from 2014 to 2018.

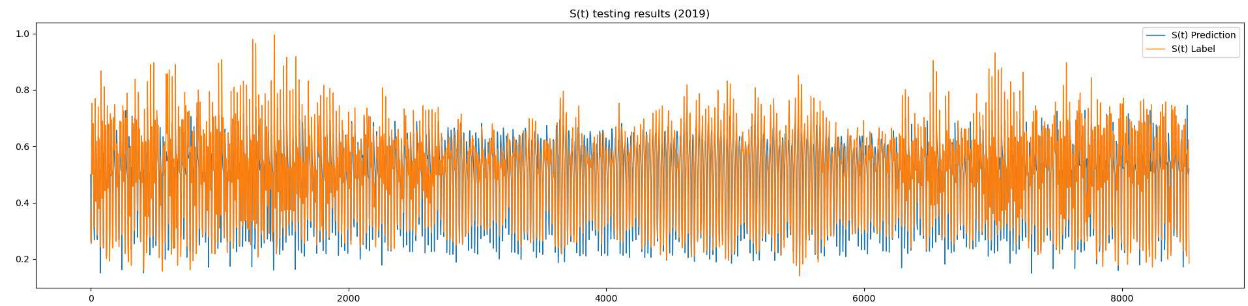

Figure 4.22: The normalized testing results of seasonal-load for BPA 2019.

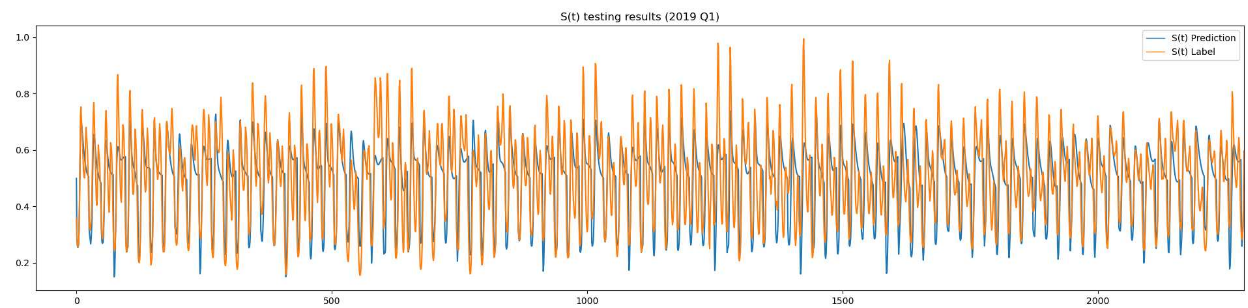

Figure 4.23: The normalized testing results of seasonal-load for BPA first quarter of 2019.

Figure 4.21 shows the training results of the normalized seasonal-load for BPA from 2014 to 2018. From the training results, we can see that our prediction of seasonal-load is in line with the changing trend of the seasonal-load label curve. Figure 4.22 shows the testing results of the normalized seasonal-load for BPA 2019, and Figure 4.23 is the zoom-in figure to show the details of the first quarter of 2019. From the zoom-in figure, we can see that our model fits well for the lower part of the seasonal-load but not so well for the higher part of the seasonal-load.

Assembled Aggregated Load Forecasting

After we obtained the forecasting load curve for both base-load and seasonal-load, the final aggregated load forecasting for BPA 2019 can be generated by assembling them.

Figure 4.24 shows the forecasting results of BPA for the entire 2019, and Figure 4.25 is the zoom-in figure to show the details of the first quarter of 2019. From these two figures, we can see that our approach has a good prediction performance.

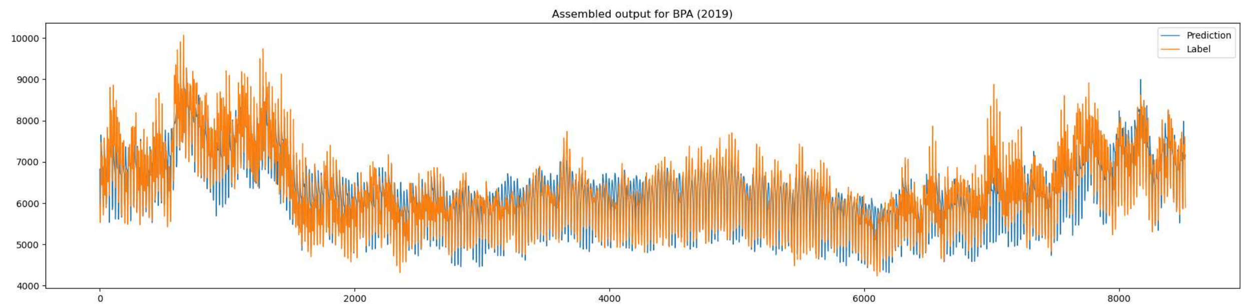

Figure 4.24: The final assembled output load curve for BPA 2019.

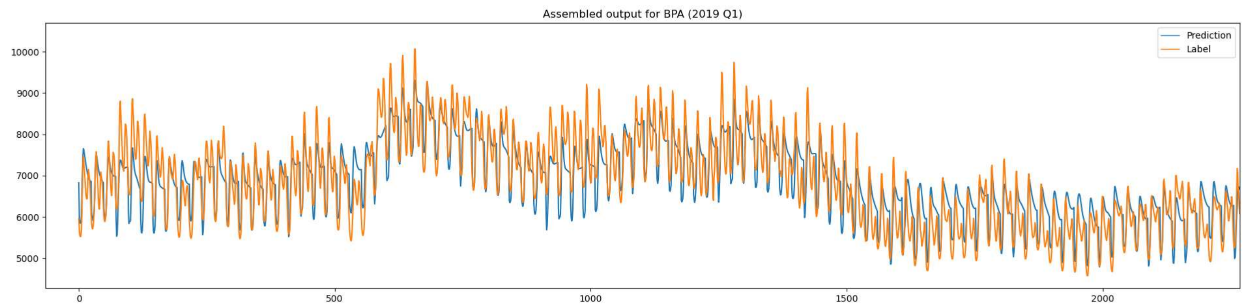

Figure 4.25: The zoom-in assembled output load curve for BPA first quarter of 2019.

Figure 4.26 shows the zoom-in figure of the week that peak load occurs, and Figure 4.27 shows the zoom-in figure of the few days that peak load occurs. These two figures demonstrate that our approach can accurately predict the time the peak load occurs for BPA.

4.2.3 COVID-19 Impact Discussion for CAISO & BPA

Due to the pandemic of COVID-19 in 2020, people are locked down in their homes as much as possible, which has changed people's daily lifestyles to a great extent. It directly impacted the load

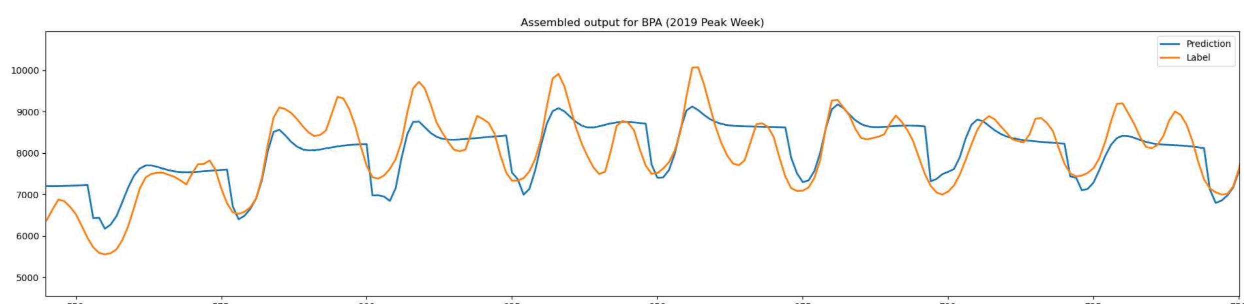

Figure 4.26: The zoom-in assembled output load curve for BPA peak load week of 2019.

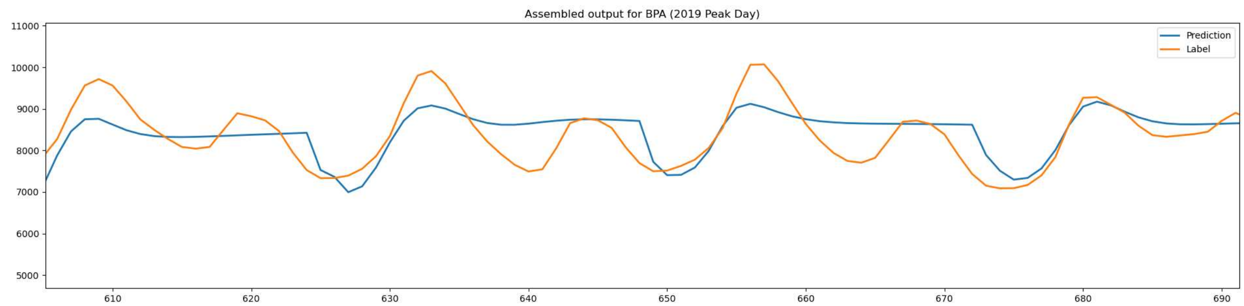

Figure 4.27: The zoom-in assembled output load curve for BPA peak load day of 2019.

profile of the power system. The entire dataset we used for CAISO and BPA includes the hourly load data from 2014 to 2020. The above case studies just presented the forecasting 2019 because our approach is to learn the load profile from previous records, which does not apply to pandemic situations. However, we can still compute the forecasting results for 2020 to see how much impact the COVID-19 brings to us.

Based on our dataset, we can predict the load curve from 2017 to 2020 using all data from 2014 to the previous year. For example, the prediction of 2017 uses a training set from 2014 to 2016, and the prediction of 2020 uses a training set from 2014 to 2019. Figure 4.28 records the normalized RMSE between the prediction curve and the label curve from 2017 to 2020. Generally speaking, the prediction error should decrease with the amount of training set increase. The normalized RMSE from 2017 to 2019 explains it. However, when we compute the prediction for 2020 with six years of data training, the normalized RMSE increased for CAISO and BPA. It is because COVID-19 changed people's daily lifestyles.

From Figure 4.28 we can also notice that the increased prediction error for 2020 of CAISO is much larger than the increased prediction error for 2020 of BPA. It demonstrates that COVID-19 brings much more impact on CAISO than on BPA. We believe that this is because of the different high load demand periods between CAISO and BPA. The high load demand period for BPA is winter around January, but the high load demand period for CAISO is summer around August. It is well known that COVID-19 broke out in North America after March and reached a dire situation in August. Therefore, COVID-19 brings much more influence to CAISO.

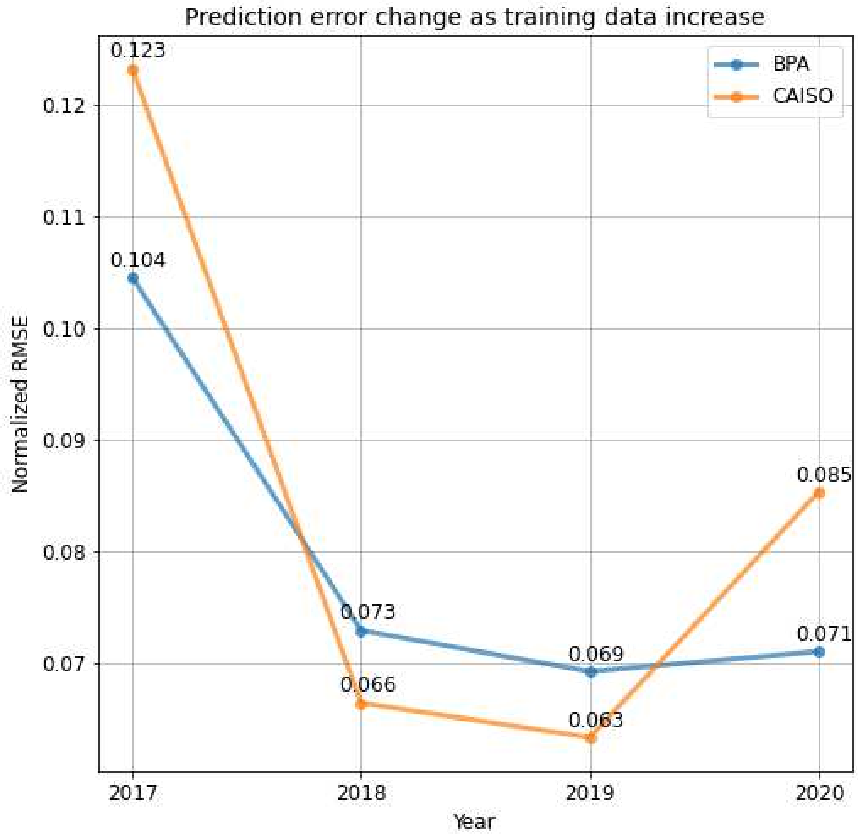

Figure 4.28: The prediction error change as training data increase for 2017 - 2020 to present the impact of COVID-19.

4.3 Summary

In this chapter, the high-resolution (hourly) aggregated load curve for a specific region in the medium-term (a few days to a few months) is generated. Because climate highly influences people's daily lifestyle, this research started from regional segmentation based on similar climatic characteristics. Then, we designed a decomposition model to decompose the hourly aggregated load curve for the region into three parts: base-load term, seasonal-load term, and the random error term. We find the relationship between the load curve and input factors by training base-load term and seasonal-load term with different learning methods (temperature and time). To highlight the temporal relationship between samples, we designed time-coding for input factors. An effective high-resolution aggregated medium-term load curve can be obtained by implementing the proposed method in this research. It can realize the accurate prediction of peak load but also the depiction of hourly load profile. It can also provide a reasonable and controllable load curve for the real-time simulation system as input. Finally, two case studies for CAISO and Bonneville Power Administration (BPA) load demand forecasting are presented. It predicted the accurate date of the peak load and the precise outline of the hourly load profile. The error of time of the predicted peak load occurs smaller than 1 hour. Eventually, by analyzing the load forecasting results for the 2020 year of CAISO and BPA, we discussed the impact of the COVID-19 on people's daily lifestyles. The results show that COVID-19 changed people's daily lifestyle a lot, and it brings much more influence on CAISO than on BPA.

Chapter 5

Conclusions

This dissertation focuses on modeling power systems with data-driven approaches, with applications in power system cyber-attack detection and recovery, and large-scale, long-term load characterization.

In the first part, the detection and recovery of availability attacks and integrity attacks in cybersecurity of the smart grid are investigated. We considered a new type of FDI named historical normal data injection. To improve the accuracy of fitting the PMU data of power systems, this research model the spatial-temporal relationship among data collected across the entire system. To ensure the capability of capturing the non-linearity caused by growing renewable energy resources, this research adopts the B-spline approximation to estimate the model. To ensure the algorithm's efficiency, this research designs an online anomaly detection approach based on the estimated model. This research demonstrates the applicability of the proposed method in this research through the simulations of ambient data of the minniWECC system. The fitting accuracy can explain the recovery capability of the proposed approach to the DoS attacked bus, using the data from remaining secure buses. By simulating the DoS attack, this research demonstrated the detection and recovery performance of the proposed method when a DoS attack occurs on a single bus and explains the impact of the number of attacked buses on the overall performance of the proposed method. This research uses historical normal data as false data to simulate the FDI attack and demonstrates the detection performance when FDI occurs on buses of a power system. In the future, we will consider the following issues: the recovery from FDI attack, the pattern analysis of the residual of each bus to distinguish different events in the power system, and the improvement of the model to implement the prediction on each bus in the power system.

In the second part, to solve the problem of non-conforming load classification, it first reduces the original data set by using the FPCA method. Then considering the complex structure and vague nature of modern loads, this article proposes two classification processors: 1) the segmental dataset processor. 2) the full dataset processor. The former can perform clustering analysis for the different characteristics of all individual loads in different periods. In displaying the results, the non-conforming characteristics of the two time periods are used to divide individual loads into four categories, and the differences between the different categories are shown. Compared to the mean function, we could give some suggestions for these four categories. $(0, 0)$ means industrial loads that are keeping non-conforming property all the time. $(0, 1)$ means some night-working residential loads that only have non-conforming property at nighttime. $(1, 0)$ means some DER users will use energy from the rooftop solar panels during the daytime, and therefore daytime, they will have a lower load measurement by the control center. $(1, 1)$ means the traditional residential loads. The latter extracts and quantifies the degree of the non-conforming component of each load based on all individual loads' performance over the entire period. The final classification result can be divided into two parts from the sorted sequence through the experience of experts. The experts in WECC could adjust the threshold according to the extent of non-conforming characteristics of loads they would concern. Some identified non-conforming loads were confirmed by the experience of experts, and other identified non-conforming loads were verified by checking their time series.

In the third part, the high-resolution (hourly) aggregated load curve for a specific region in the medium-term (a few days to a few months) is generated. Because climate highly influences people's daily lifestyle, this research started from regional segmentation based on similar climatic characteristics. Then, we designed a decomposition model to decompose the hourly aggregated load curve for the region into three parts: base-load term, seasonal-load term, and the random error term. We find the relationship between the load curve and input factors by training base-load term and seasonal-load term with different learning methods (temperature and time). To highlight the temporal relationship between samples, we designed time-coding for input factors. An effective high-resolution aggregated medium-term load curve can be obtained by implementing the proposed method in this research. It can realize the accurate prediction of peak load and the depiction of hourly load profile. It can also provide a reasonable and controllable load curve for the real-time simulation system as input. Finally, two case studies for CAISO and BPA load demand forecasting are presented. It predicted the accurate date of the peak load and the precise outline of the hourly load profile.

Bibliography

[1] Colah . Understanding LSTM Networks . https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Online;.

[2] C. Ten, G. Manimaran, and C. Liu. Cybersecurity for critical infrastructures: Attack and defense modeling. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(4):853–865, 2010.

[3] Ahmed S Musleh, Guo Chen, and Zhao Yang Dong. A survey on the detection algorithms for false data injection attacks in smart grids. *IEEE Transactions on Smart Grid*, 11(3):2218–2234, 2019.

[4] Department of Energy. Grid Modernization and the Smart Grid. https://www.energy.gov/oe/activities/technology-development/grid-modernization-and-smart-grid. Online;.

[5] Ataul Bari, Jin Jiang, Walid Saad, and Arunita Jaekel. Challenges in the smart grid applications: an overview. *International Journal of Distributed Sensor Networks*, 10(2):974682, 2014.

[6] Shitharth Selvarajan, Masood Shaik, Sirajudeen Ameerjohn, and Sangeetha Kannan. Mining of intrusion attack in scada network using clustering and genetically seeded flora-based optimal classification algorithm. *IET Information Security*, 14(1):1–11, 2019.

[7] Yacine Chakhchoukh, Hangtian Lei, and Brian K Johnson. Diagnosis of outliers and cyber attacks in dynamic pmu-based power state estimation. *IEEE Transactions on Power Systems*, 35(2):1188–1197, 2019.

[8] Mohammad Esmalifalak, Lanchao Liu, Nam Nguyen, Rong Zheng, and Zhu Han. Detecting stealthy false data injection using machine learning in smart grid. *IEEE Systems Journal*, 11(3):1644–1652, 2014.

[9] Power Technology. The five worst cyberattacks against the power industry since 2014. https://www.power-technology.com/features/the-five-worst-cyberattacks-against-the-power-industry-since2014/. Online;.

[10] Raymond C Borges Hink, Justin M Beaver, Mark A Buckner, Tommy Morris, Uttam Adhikari, and Shengyi Pan. Machine learning for power system disturbance and cyber-attack discrimination. In *2014 7th International symposium on resilient control systems (ISRCS)*, pages 1–8. IEEE, 2014.

[11] S. Pan, T. Morris, and U. Adhikari. Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data. *IEEE Transactions on Industrial Informatics*, 11(3):650–662, 2015.

[12] Haicheng Tu, Yongxiang Xia, Chi K. Tse, and Xi Chen. A hybrid cyber attack model for cyber-physical power systems. *IEEE Access*, 8:114876–114883, 2020.

[13] Md Abul Hasnat and Mahshid Rahnamay-Naeini. A data-driven dynamic state estimation for smart grids under dos attack using state correlations. In *2019 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2019.

[14] Dou An, Qingyu Yang, Wenmao Liu, and Yang Zhang. Defending against data integrity attacks in smart grid: A deep reinforcement learning-based approach. *IEEE Access*, 7:110835–110845, 2019.

[15] Xinyu Wang, Xiaoyuan Luo, Yuyan Zhang, and Xinping Guan. Detection and isolation of false data injection attacks in smart grids via nonlinear interval observer. *IEEE Internet of Things Journal*, 6(4):6498–6512, 2019.

[16] Xinyu Wang, Xiaoyuan Luo, Yuyan Zhang, and Xinping Guan. Detection and isolation of false data injection attacks in smart grids via nonlinear interval observer. *IEEE Internet of Things Journal*, 6(4):6498–6512, 2019.

[17] Kamal Jain. Security based on network topology against the wiretapping attack. *IEEE Wireless Communications*, 11(1):68–71, 2004.

[18] Charles V Wright, Scott E Coull, and Fabian Monrose. Traffic morphing: An efficient defense against statistical traffic analysis. In *NDSS*, volume 9. Citeseer, 2009.

[19] Chee-Wooi Ten, Junho Hong, and Chen-Ching Liu. Anomaly detection for cybersecurity of the substations. *IEEE Transactions on Smart Grid*, 2(4):865–873, 2011.

[20] Maurilio P Coutinho, Germano Lambert-Torres, LE Borges da Silva, HG Martins, Horst Lazarek, and J Cabral Neto. Anomaly detection in power system control center critical infrastructures using rough classification algorithm. In *2009 3rd IEEE International Conference on Digital Ecosystems and Technologies*, pages 733–738. IEEE, 2009.

[21] M. A. Rahman, E. Al-Shaer, and R. G. Kavasseri. A formal model for verifying the impact of stealthy attacks on optimal power flow in power grids. In *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, pages 175–186, 2014.

[22] Aditya Ashok, Manimaran Govindarasu, and Venkataramana Ajjarapu. Online detection of stealthy false data injection attacks in power system state estimation. *IEEE Transactions on Smart Grid*, 9(3):1636–1646, 2016.

[23] Jorge Valenzuela, Jianhui Wang, and Nancy Bissinger. Real-time intrusion detection in power system operations. *IEEE Transactions on Power Systems*, 28(2):1052–1062, 2012.

[24] Shengyi Pan, Thomas Morris, and Uttam Adhikari. Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, 6(6):3104–3113, 2015.

[25] I. Lukicheva, D. Pozo, and A. Kulikov. Cyberattack detection in intelligent grids using non-linear filtering. In *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, 2018.

[26] G. Shen, L. Jiaying, W. Peng, Z. Jichuan, and L. Ce. Bad data identification in power system based on optimized gsa algorithm. In *2019 Chinese Control And Decision Conference (CCDC)*, pages 5763–5767, 2019.

[27] Michael Starke, Nasr Alkadi, and Ookie Ma. Assessment of industrial load for demand response across u.s. regions of the western interconnect. 9 2013.

[28] G. J. Tsekouras, N. D. Hatziargyriou, and E. N. Dialynas. Two-stage pattern recognition of load curves for classification of electricity customers. *IEEE Transactions on Power Systems*, 22(3):1120–1128, 2007.

[29] QIAN Tiantian, WANG Ke, SHI Fei, and ZHU Kedong. Analysis of characteristics of industrial power load in a certain region. *IOP Conference Series: Materials Science and Engineering*, 563(5):052109, aug 2019.

[30] L. D. Douglas, T. A. Green, and R. A. Kramer. New approaches to the agc nonconforming load problem. In *Conference Proceedings Power Industry Computer Application Conference*, pages 48–57, 1993.

[31] J. O. Ramsay. *Functional Data Analysis*. American Cancer Society, 2006.

[32] K.B. Lindberg, P. Seljom, H. Madsen, D. Fischer, and M. Korpås. Long-term electricity load forecasting: Current and future trends. *Utilities Policy*, 58:102–119, 2019.

[33] N Phuangpornpitak and W Prommee. A study of load demand forecasting models in electric power system operation and planning. *GMSARN International Journal*, 10(2016):19–24, 2016.

[34] J Hartono, AA Surya, RS Utami, Bbsda Harsono, Hb Tambunan, and Ap Purnomoadi. Long term load demand forecasting in bali province using deep learning neural network. In *2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP)*, pages 174–178, 2020.

[35] Zhen Shao, Shan-Lin Yang, and Fei Gao. Density prediction and dimensionality reduction of mid-term electricity demand in china: A new semiparametric-based additive model. *Energy Conversion and Management*, 87:439–454, 2014.

[36] Juan Pablo Carvallo, Peter H. Larsen, Alan H. Sanstad, and Charles A. Goldman. Long term load forecasting accuracy in electric utility integrated resource planning. *Energy Policy*, 119:410–422, 2018.

[37] Netzentwicklungsplan Strom. Zweiter entwurf der übertragungsnetzbetreiber, 50hertz transmission gmbh. *Amprion GmbH, TenneT TSO GmbH, TransnetBW GmbH, Berlin*, 2014.

[38] Kianoosh G. Boroojeni, M. Hadi Amini, Shahab Bahrami, S.S. Iyengar, Arif I. Sarwat, and Orkun Karabasoglu. A novel multi-time-scale modeling for electric power demand forecasting: From short-term to medium-term horizon. *Electric Power Systems Research*, 142:58–73, 2017.

[39] Yannig Goude, Raphael Nedellec, and Nicolas Kong. Local short and middle term electricity load forecasting with semi-parametric additive models. *IEEE Transactions on Smart Grid*, 5(1):440–446, 2014.

[40] Gobind G. Pillai, Ghanim A. Putrus, and Nicola M. Pearsall. Generation of synthetic benchmark electrical load profiles using publicly available load and weather data. *International Journal of Electrical Power & Energy Systems*, 61:1–10, 2014.

[41] Intisar A. Sajjad, Gianfranco Chicco, and Roberto Napoli. Effect of aggregation level and sampling time on load variation profile — a statistical analysis. In *MELECON 2014 - 2014 17th IEEE Mediterranean Electrotechnical Conference*, pages 208–212, 2014.

[42] Matteo De Felice, Andrea Alessandri, and Franco Catalano. Seasonal climate forecasts for medium-term electricity demand forecasting. *Applied Energy*, 137:435–444, 2015.

[43] H.M. Al-Hamadi and S.A. Soliman. Long-term/mid-term electric load forecasting based on short-term correlation and annual growth. *Electric Power Systems Research*, 74(3):353–361, 2005.

[44] F.M. Andersen, H.V. Larsen, and T.K. Boomsma. Long-term forecasting of hourly electricity load: Identification of consumption profiles and segmentation of customers. *Energy Conversion and Management*, 68:244–252, 2013.

[45] Pituk Bunnoon, Kusumal Chalermyanont, and Chusak Limsakul. Mid-term load forecasting: Level suitably of wavelet and neural network based on factor selection. *Energy Procedia*, 14:438–444, 2012. 2011 2nd International Conference on Advances in Energy Engineering (ICAEE).

[46] Tamer Ahmed Farrag and Ehab E. Elattar. Optimized deep stacked long short-term memory network for long-term load forecasting. *IEEE Access*, 9:68511–68522, 2021.

[47] Vito Volterra. Theory of functionals and of integral and integro-differential equations. 1959.

[48] Ela Sienkiewicz, Dong Song, F. Jay Breidt, and Haonan Wang. Sparse functional dynamical modelsâ a big data approach. *Journal of Computational and Graphical Statistics*, 26(2):319–329, 2017.

[49] William J Gordon and Richard F Riesenfeld. B-spline curves and surfaces. In *Computer aided geometric design*, pages 95–126. Elsevier, 1974.

[50] Lori A. Dalton and Edward R. Dougherty. Bayesian minimum mean-square error estimation for classification error—part i: Definition and the bayesian mmse error estimator for discrete classification. *IEEE Transactions on Signal Processing*, 59(1):115–129, 2011.

[51] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003.

[52] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[53] Robert R Meyer and Philip M Roth. Modified damped least squares: an algorithm for non-linear estimation. *IMA Journal of Applied Mathematics*, 9(2):218–233, 1972.

[54] Nathan Mantel. Chi-square tests with one degree of freedom; extensions of the mantel-haenszel procedure. *Journal of the American Statistical Association*, 58(303):690–700, 1963.

[55] D Trudnowski and J Undrill. The minniwecc system model. *Oscillation damping controls*, 6:8, 2008.

[56] Robert GD Steel. Principles and procedures of statistics: with special reference to the biological sciences. Technical report, 1960.

[57] Kai le Zhou, Shan lin Yang, and Chao Shen. A review of electric load classification in smart grid environment. *Renewable and Sustainable Energy Reviews*, 24:103 – 110, 2013.

[58] H. Yang, J. Zhang, J. Qiu, S. Zhang, M. Lai, and Z. Y. Dong. A practical pricing approach to smart grid demand response based on load classification. *IEEE Transactions on Smart Grid*, 9(1):179–190, 2018.

[59] Mathieu Bourdeau, Philippe Basset, Solène Beauchêne, David Da Silva, Thierry Guiot, David Werner, and Elyes Nefzaoui. Classification of daily electric load profiles of non-residential buildings. *Energy and Buildings*, 233:110670, 2021.

[60] Western Electricity Coordinating Council (WECC) . WECC Modeling and Validation Work Group Load Long ID (LID) Instructions. https://www.wecc.org/Reliability/LID_Instructions_2012-4-10.pdf. Online;.

[61] Han Lin Shang. A survey of functional principal component analysis. *AStA Advances in Statistical Analysis*, 98(2):121–142, Apr 2014.

[62] Hervé Abdi and Lynne J. Williams. Principal component analysis. *WIREs Computational Statistics*, 2(4):433–459, 2010.

[63] Henry F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, Sep 1958.

[64] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[65] B. A. Kimball. Smoothing data with fourier transformations1. *Agronomy Journal*, 66(2):259–262, 1974.

[66] Tony D. DeRose and Brian A. Barsky. Geometric continuity, shape parameters, and geometric constructions for catmull-rom splines. *ACM Trans. Graph.*, 7(1):1–41, January 1988.

[67] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[68] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 5(4):455–455, 1992.

[69] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[70] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[71] Jason Brownlee . How to Check if Time Series Data is Stationary with Python . https://machinelearningmastery.com/time-series-data-stationary-python/. Online;.

[72] CAISO . Current and forecasted demand for CAISO . https://www.caiso.com/TodaysOutlook/Pages/default.aspx. Online;.

[73] NOAA . Climate Data Online: Dataset Discovery . https://www.ncdc.noaa.gov/cdo-web/datasets. Online;.

[74] BPA . Bonneville Power Administration: BPA.gov . https://www.bpa.gov/Pages/home.aspx. Online;.