Summer 7-31-2019

A Machine Learning Based Framework for Load Forecasting And Optimal Operation of Power Systems with Distributed Generation

Tairen Chen

Tairen Chen
candidate

Electrical and Computer Engineering
Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Jane M. Lehr, Chairperson

Dr. Manel Martínez-Ramón

Dr. Olga Lavrova

Dr. Andrea A. Mammoli

A Machine Learning Based Framework for Load Forecasting And Optimal Operation of Power Systems with Distributed Generation

by

Tairen Chen

B.E., Automation, Central South University, 2005
M.S., Electrical Engineering , Central South University, 2008

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2017

Dedication

I dedicate this work to God, for everything He gave and will give to me.

To my mother, Liping, my father, Shian, and my wife, Ao, for their support and encouragement.

"Continuous effort–not strength or intelligence–is the key to unlocking our potential."

— Winston Churchill

Acknowledgments

First, I wholeheartedly thank my adviser, Dr. Jane Lehr. Without her support and guidance, it is impossible for me to finish the Ph.D. studies. For these years in the UNM, she has taught me from how to do research to how to build my career. She is my closest friend in the UNM. With equal thank to my co-adviser Dr. Manel Martínez-Ramón. He introduced me to the machine learning world, and he has directed and corrected my work with great wisdom and patience. I should thank Dr. Olga Lavrova. She introduced me to the smart grid and helped me when I was in the darkest time of my life. I would also like to thank Dr. Andrea A. Mammoli for his help and support on my research and thesis.

I would like to thank the Electrical and Computer Engineering Department and the Center for Advanced Research Computing. They provide me with the funding and the computing resource, which help me to finish my dissertation. I would also like to thank the faculty, lab members, my host family and other people that used to help me, more or less, during the time that I have studied at the UNM.

In the last, I would like to thank my parents and my wife, who support, encourage, and pray for me in every situation. With their love, my life journey of pursuing a PhD degree is becoming colorful and meaningful.

A Machine Learning Based Framework for Load Forecasting And Optimal Operation of Power Systems with Distributed Generation

by

Tairen Chen

B.E., Automation, Central South University, 2005
M.S., Electrical Engineering , Central South University, 2008

Ph.D., Engineering, University of New Mexico, 2017

Abstract

The fast development and wide utilization of distributed generations (DGs), such as Photovoltaic panels and wind turbines, provide environmentally friendly renewable energy. However, inappropriate operation, sizing, and placement of DGs could increase the power losses and reduce the stability of the power network. Load forecasting is critical to the electrical utilities to schedule power generation and distribution. In this dissertation, a framework is proposed for load forecasting and optimal operation of power system with DGs in the distribution feeder-level.

In the first part, a nonparametric method, the Bayesian Additive Regression Trees (BART), is introduced for day-ahead peak load forecasting. The detailed correlation analysis of peak load and weather information is performed in a residential area in Albuquerque and a business area in the North Central of Texas for two different two-years periods. Next, the BART method is applied with a principled permutation-based inferential variable selection approach. The BART method's prediction accuracy is then compared with the Multiple Linear Regression (MLR), the Support Vector Machine (SVM) and the composite kernel of Gaussian Process Regression (GPR).

The forecasting results are then measured by Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Forecasting Error (MAFE), R^2, and Mean Absolute Percentage Error (MAPE). The BART method displays the best prediction accuracy for every index.

In the second part, a new framework of distribution feeder-level short-term and very short-term load forecasting is proposed. First, a composite Matérn kernels (CMKs) based Gaussian Process is designed for day-ahead load forecasting based on four years of recorded data and kernels comparison. A data selection algorithm is proposed to improve the prediction further. Second, a three-step daily curve tuning algorithm is designed based on the dictionary learning algorithm, K-SVD, to improve the forecasting results further. In step one, the dictionary is built by using the K-SVD to decompose the output of the CMKs. In step two, for a certain length of atoms, tuned curves are generated by using the K-SVD to learn the known daily load. A curve selection model is designed to choose the best-tuned curve based on the linear regression models with forecasting errors as feedback. In step three, the final tuned curve is selected by the minimization of the mean daily load difference. The framework is verified using two-year private data from the residential area and two-year public data from the business area with three aspects of results.

In the third part, an optimal method to plan and dynamically operate the DG based on the modified nondominated sorting genetic algorithm II (NSGA-II) is proposed. First, the uncertainty of load and DG (photovoltaic panels) output are considered. Second, the placement of a DG is defined by voltage sensitivity analysis. A multi-objective problem is then formulated to find the optimal daily operation of DG. To solve the problem, a fuzzy logic decision model is designed to modify the traditional NSGA II that selects an optimally compromised solution from the Pareto front. Furthermore, to increase the modified NSGA II computation speed, the population initialization space is reduced, and the population is selected and saved for the next generation based on load analysis. With the accurate load forecasting as in part one and two, the initialization space could be reduced further. The method is tested on the IEEE 14 bus, and it is compared with other two optimal methods. The results on reducing the power losses, voltage deviations, and increasing the algorithm speed demonstrate the effectiveness of this method.

Contents

List of Figures

List of Tables

Glossary

DG/DGs	Distributed Generation / Distributed Generations.
PV	Photovoltaic
DGPS	Distributed Generation Placement and Sizing.
DR	Demand Response
STLF	Short-Term Load Forecasting
VSTLF	very short-term load forecasting
SVM	Support Vector Machine
SVR	Support Vector Regression
GP	Gaussian Process
GPR	Gaussian Process Regression
LR	Linear Regression
ARMA	Autoregressive moving average
GA	Genetic Algorithm
SA	Simulated annealing
PSO	Particle Swarm Optimization

ERCOT	The Electric Reliability Council of Texas
EV	Electric Vehicle
BART	Bayesian Additive Regression Trees
MLR	Multiple Linear Regression
MSE	Mean Square Error
RMSE	Root Mean Square Error
MAFE	Mean absolute Forecasting Error
ARDSE	Automatic relevance determination squared exponential
CMKs	Composite Matérn Kernels
ORMP	Order Recursive Matching Pursuit algorithm
PMK	Product of Matérn Kernels
AMK	Additive of Matérn Kernels
MMLDR	Monthly mean load demand ratio
YMAPE	Yearly mean absolute percent error
SG	Savitzky-Golay filter
DLA	Dictionary Learning Algorithm
TP/HTP	time point / Historical time point
CTA	Curve Tuning Algorithm
DAMSEY	Daily average MSE for each year
DGTMSE	Daily gradually tuning MSE

NSGA II	nondominated sorting genetic algorithm
MOO	multiobjective optimization
$\|\| \cdot \|\|_F$	the Frobenius norm
$\circ$	Hadamard mutiplication
$\oslash$	Hadamard division
$\overline{\mathbf{x}}$	the mean of vector $\mathbf{x}$
$\overline{\mathbf{X}}$	the column mean of matrix $\mathbf{X}$
diag$(d_1, \cdots, d_N)$	A diagonal matrix with $(d_1, \cdots, d_N)$ on its main diagonal.

Chapter 1

Introduction

1.1 Background

During hundreds of years of development at power supply and distribution, central-station generators continuously play a major role in the production of power. Although this system can provide relatively inexpensive power, issues remain, such as increasing fuel costs, reducing power plant emissions, and increasing customer needs for higher reliability power [1]. One of promising solutions is to employ a decentralized system composed of small generation systems on the distribution network. This type of generation system is known as Distributed Generation (DG) [2]. DGs usually are interconnected near the load in the power distribution network rather than near the bulk power distribution system. The typical individual DG unit rating is less than 10 MVA and includes fossil fuel and renewable generations, as well as energy storage technologies. Appropriately utilized DGs are efficient than central-station generators because they are closer to the customer load, which means there is less distribution and transmission loss. The application of DGs on distribution networks could reduce the system's power losses, improve the voltage profile, increase the network reliability, and defer network expansion, etc. [3–5]

In recent years, increased use worldwide of distributed energy, such as photovoltaic (PV), wind

turbine, and fuel cell, has gained great acceptance because such energies are renewable and environmentally friendly. However, due to misplacement, inappropriate sizing, and operation of DGs, they also pose the challenge to the distribution network operator because a high penetration of DG in certain places can result in voltage deviation and power losses [6, 7]. To tackle these problems, methods have been advanced to handle the placement and sizing of DGs. In general, methods can be divided into three types: analytical method, numerical method, and heuristic method [8]. Two types of objective functions are incorporated: single objective function is formulated by minimization of power loss [9–12], and multi-objective optimization problems are formulated by minimization the power loss, voltage deviation, and profiles, etc [13–16]. Among these categories, the heuristic method for multiobjective optimization is robust and works well for large and complex DG planning problems [17].

However, there are several issues that need to be further concerned about DGs Placement and Sizing (DGPS).

First, with DGPS, most current methods address only the static cases, in which the load is considered to be unchanged or to be a one-level load. In actuality, loads change nearly every second. Though there are few references, such as [18, 19], discussing the variant load, they do not consider the PV as DGs, which is widely used DG around world. Second is DGs operation. After DGs are installed, their optimal operation is crucial for utility companies. Since DGs do not continuously work at one-level output, such as, maximum or minimum, the dynamic operation and uncertainty of DGs with variant output need to be considered during daily circuits operation in order to save losses of energy. Last but not least, to operate the DGs optimally, accurate forecasting of the load demand is crucial for the utilities. The accurate load demand forecasting, especially the short-term and very short-term, not only provide the utilities advantages at the trading action in the electricity market, but also afford a timing advantages to optimally allocate the power and operate the DGs.

To address these issues, load forecasting and the DGPS are required to be further studied, and inter-disciplinary designs and strategies should be considered from areas such as machine learning, evolutionary computation and optimization, and control theory.

In this thesis, we work on building the peak load forecasting, Very Short-term Load Forecasting (VSTLF) and the Short-term Load Forecasting (STLF) models, which include the load diversity and uncertainty, to predict the load for DGPS, designing a dynamic DGPS and DGs' operation strategy, and analyzing circuit network stability. Before presenting details of the methods used, we provide a short survey of recent works for the following topics: machine learning methods for load forecasting and optimization methods for DGPS.

1.2 Literature Review

1.2.1 Short-term and Very Short-term Load Forecasting

For the electric utility companies, there are several critical issues regarding the smart grids. These issues include managing demand response (DR) to reduce peak electricity load impact, design an economic plan to generate and/or purchase power and to distribute the power optimally. By solving these issues, the utilities can delay and save future investment in the power generation and transmission, and better utilize renewable energies to reduce our dependence on hydrocarbon. Load forecasting is one of the important factors that DR depends on [20]. Among the different length of load forecasting horizon, the VSTLF and STLF received a lot of attention from both the academic and industrial groups, because the forecasting quality is critical to the smart grid operation and the transaction activities of the competitive electricity market. The VSTLF models, with forecasting horizon from seconds or minutes to one hour, are usually used to control the power flow. The STLF models, from hours to weeks, are generally used to adjust generation and demand. [21]. But accurate VSTLF and STLF are difficult because loads not only have the properties of circularity and seasonality, but also are sensitive to several factors, such as weather, human activities, and circuit level. Generally speaking, to do the STLF and VSTLF, there are three types of machine learning related methods: traditional methods, neural network methods, Kernel methods, such as Support Vector Machine (SVM), Gaussian Process Regression (GPR), and their hybrid methods.

One type of traditional methods is the regression model, which includes linear regression (LR), multiple LR, and dynamic LR. They usually utilize weather data as key explanatory to forecast load because of the strong linear relationship between temperature and load [22–24]. The autoregressive moving average (ARMA) and the autoregressive integrated moving average (ARIMA) also have been used in load demand forecasting [25–28]. The Kalman filter-based methods are widely used for load forecasting. Kalman is a statistical technique which provides information about the quality of the estimation; this technique provides, in addition to the best estimate, the variance of the estimation error for linear and Gaussian models. [29–32]. More methods can be found in [20].

Load and weather data are not always in a linear relationship; sometimes the data displays a nonlinear relationship between them, since human activities, such as festivals and events, also alters the load. Neural network-based methods, in such a case, provide a powerful tool to fit the nonlinear properties.

Neural network methods, a branch of artificial intelligence (AI), are also popular with applications at the VSTLF and the STLF. Different types of neural networks are used for forecasting. In [33], a forecasting model is established by combining the radial basis function (RBF) neural network with the adaptive neural fuzzy inference system. In [34], they present an approach of back propagation neural network with a rough data set for complicated STLF with dynamic and nonlinear factors to develop the accuracy of predictions. [35,36], the multilayer perceptron (MLP), which is a type of artificial neural network, is used for STLF with real load and weather data. They further investigate an efficient way to use the MLP neural network at STLF, a way that does not require the history data. A similar day-based wavelet neural network method to forecast next day's load is presented at [37]. These authors select similar day load as the input load based on correlation analysis, and use wavelet decomposition and separate neural networks to capture the features of load at low and high frequencies. Additional methods are included in [38, 39].

The SVM and the GPR are the learning method that is theoretically built on statistical learning theory. The SVM was originally developed for classification and later generalized to solve regression problems [40–43]. The generalized method is also called support vector regression (SVR). SVR has been successfully applied to the load forecasting area, but with large amounts of data, its

computing speed is significantly slow. In [44], before applying SVR, they take advantage of data mining technology to process large data and remove redundant information. The same strategy is utilized to search historical load and meteorological data for the input of SVR, which substantially increases the SVR computation speed. The GPR is another popular kernel method that is equipped with probabilistic property and the uncertainty estimation [97]. In [113] and [114], the single common kernels are introduced and compared to do load forecasting, which display advantages than the multiple LR and the SVR. In [115], a twin Gaussian Process, placing the Gaussian priors on both covariance and responses, acquires the output via Kullback-Leibler divergence minimization between two GP modeled as normal distributions over finite index sets of training and testing examples. This method shows that twin Gaussian Process can be a useful tool for load forecasting.

There are hybrid kernel methods that are utilized to do load forecasting. To determine the correct parameter set for selected STLF, several researchers use the evolutionary techniques to work out the parameters of SVR, such as chaotic artificial bee colony algorithm, chaotic particle swarm optimization algorithm, and differential evolution algorithm at [45–47]. Also, there are hybrid methods designed by combining traditional methods, like Kalman filter, with SVR as the new kernel, which could compare with Extended Kalman Filters and Unscented Kalman Filters [48,49]. To further investigate the kernel, [50] proposes a kernel-based SVR combination model by using a novel algorithm for individual model selection. Moreover, the proposed combination model provides a new way to select kernel function for SVR model. And this model demonstrates better forecasting result than the best individual kernel-based SVR model. More descriptions of SVR related methods can be found in [51]. There are few hybrid GP method too. The ensemble GPR sub-models are selected by the genetic algorithm to do very short-term load forecasting in [118]. In [117], five GPR sub-models are integrated by a neural network to do wind power forecasting.

Other methods, including hybrid method built with a traditional method and evolutionary techniques, artificial intelligence method with evolutionary method, Semi Parametric additive model, and grey correlation contest model can be found in [52–55].

1.2.2 Optimization for Distributed Generation Placement and Sizing

Optimal placement and sizing of DG is very important to the utility and environment, since it saves power losses, increases reliability of the power network, and defers the system upgrading, as we mentioned before. Here, we introduce the methods that are currently available for DGPS in detail. In general, we can summarize current available methods into three types: the traditional methods, evolutionary computation methods, and hybrid methods [17].

The traditional methods mainly include analytical methods and numerical methods. Various analytical methods have been proposed for the DGPS. Most of these methods are built on theoretical, mathematical analysis and calculation.

In [56], the 2/3 rule is used to locate the DGs to reduce the reverse power flow and remove the zero point when DG output is bigger than the load downstream of the DG location. [57] use the Kalman filter algorithm to find the DG size after they get the optimal location for the DG by analyzing power losses in a steady state. An analytical expression for finding the optimal size and power factor of four types of DG units is proposed in [9], which improves the situation in which only real power can be delivered. Based on the equivalent current injection, the authors of [58] formulate a loss sensitivity factor for the distribution system. From the loss sensitivity factor, they find optimum size and location for DG without the use of admittance matrix, inverse of admittance matrix, or Jacobian matrix. For numerical methods, [59] proposes a methodology based on linear programming to determine the optimal allocation of DG with respect to their network constraints. In [60], optimal sites for placement of DGs have been identified by using the mixed integer nonlinear programming approach, and the sizing of the DGs is also taken into consideration for increasing the system's reliability. Using the time varying load, [61] proposes an approach based on dynamic programming to solve the multi-objective problem for DGPS, which is built on cost/benefit forms. In [62], the optimal DG location is found by an exhaustive search of possible combinations, and the rating of DG is calculated by fast sequential quadratic programming, which is proposed by the authors. Also, an ordinal optimization approach is proposed by [63] to locate and size multiple DGs, such that a trade-off between loss minimization and DG capacity maximization is achieved.

While traditional methods have the properties of fast computation and easy implementation. They make simplified assumptions and consider simple power system loading situations. Heuristic methods are usually robust and can provide near optimal solution for large and complex DGPS problems [17].

Evolutionary computation is a branch of heuristic methods. Evolutionary computation is a branch of artificial intelligence. It is composed of a type of algorithm that is based on adopting Darwinian evolutionary principles. There are many evolutionary computation techniques that are used for DGPS, such as the genetic algorithm (GA), simulated annealing, particle swarm optimization (PSO), differential evolution, ant colony system algorithm, artificial bee colony algorithm, Tabu search, firefly algorithm, cuckoo search and imperialist competition algorithm. In [64, 65], the authors use the GA based-method to determine the optimal location and size of the distributed generations to be placed in radial, as well as networked, systems, with the objective of minimizing the power loss. They also consider different load models. In [66], the simulated annealing technique is implemented to optimize the sizing and location of distributed generation facilities, and cost minimization is also achieved. In [67], particle swarm optimization, incorporated with the index-based approach, is used for optimally determining the size and location of multi-distributed generation units in distribution systems with different load models. They also show that the load models can significantly affect DGPS in a distribution system. [68] presents the differential evolution for DGPS in distribution networks considering various technical and economic aspects of the problem and formulates a multi-objective function. This function includes the cost of network upgrading, purchased energy and energy losses, total voltage deviation, and total capacity release. In [69], an ant colony system algorithm is used to derive the optimal recloser and DG placement scheme for radial distribution networks. A composite reliability index is used as the objective function in the optimization procedure. [70] proposes the artificial bee colony algorithm to determine the optimal DG-unit's size, power factor, and location in order to minimize the total system real power loss, which is further tested at the radial power network system. In [71, 72], Tabu search method is first used for DGPS to reduce the system losses, and further it is compared with other heuristic methods to show its efficiency. An application of firefly algorithm for DGPS is imple-

mented in [73], which is inspired by the flashing behavior of fireflies and the brightness of the signal to attract other fireflies. In [74, 75], a meta-heuristic optimization algorithm, called cuckoo search, is developed, and it is further applied to the DGPS to optimize the voltage profile and save the power losses. In [76, 77], imperialist competitive algorithm is proposed as a new sociopolitical motivated global search strategy, and it is implemented to maximize the benefits of distribution network operators for both active loss reduction and network investment deferral incentives.

Hybrid methods are defined as the combination of existing techniques, especially the combination of evolutionary computation techniques. The objective is to take advantage of different techniques to form a new one so that the new method can outweigh its components in solution accuracy and computation efficiency. A GA-Tabu search algorithm is proposed in [78], which is applied to the DGPS problem. The GA-Tabu method shows much better performance at solution accuracy and convergence process. In [14], the GA-PSO method is developed to do DGPS. The authors utilize the GA to optimize the DGs' location and use the PSO to optimize the size of the DGs. The GA-PSO displays better performance at objective function variance than individual methods that are based on GA or PSO. Also [79] develops an algorithm that combines the GA and immune systems to maximize the benefits of distribution network operators due to sizing and placement of DG units in distribution networks. In [80], a Tabu-Fuzzy method is proposed to solve the problem of DGPS. The new fuzzy model finds the nondominated multiobjective solutions corresponding to the simultaneous optimization of the fuzzy economic cost, level of fuzzy reliability, and exposure of such networks, using an original and powerful meta-heuristic algorithm based on Tabu Search. This model determines the optimal location and size of the future feeders and substations in distribution networks with dimensions significantly larger than the other methods. In [81], a new PSO is proposed by combining PSO and load flow algorithm to optimally incorporate a DG into a distribution system. It optimizes total network power losses while satisfying the voltage constraints imposed on the system, and finds the best combination of location and size simultaneously. More methods can be found in [82].

1.3 Outline of This Dissertation

To build a framework for optimal DGs placement, sizing, and operation, this research focuses on the following topics: peak load forecasting, very short-term load forecasting, short-term load forecasting, evolutionary algorithm and its application at DGs placement, sizing and operation. To optimally operate the DG in real time, it is important to have the accurate load forecasting in a time range. The load prediction in this thesis is separated into day-ahead peak load forecasting, day-ahead whole-day load prediction and very short-term load forecasting, so to help the electrical utility companies to make an economic power generation and trading schedule and optimally distribute the power.

Chapter 2 focuses on day-ahead peak load forecasting for both the residential area in Albuquerque, New Mexico and the business area in the North Central of Texas. Thorough data analysis is presented, and the Bayesian Additive Regression Trees (BART) method is introduced for day-ahead peak load forecasting. The method is further compared with the multiple linear regression, the SVM, and the composite kernel based GPR.

Chapter 3 considers the problem of whole day load prediction and VSTLF in the same areas as chapter 2. The time series load data are used for the modeling. The day-ahead STLF models are designed by utilizing the composite Matérn kernels (CMKs) based on the GPR. Date selection algorithm is designed to improve the forecasting accuracy of the CMKs. To improve the VSTLF of the CMKs, the curve tuning algorithm is designed based on the K-SVD with three aspects results.

Chapter 4 concerns the optimal DGPS and its operation analysis. A method based on the evolutionary computation algorithm, nondominated sorting genetic algorithm, is introduced, and fuzzy logic decision model is designed to choose the solution from the Pareto front. To increase the efficiency of non-sorting GA for DG operation, several techniques are proposed. The method is tested at the IEEE 14 buses and it is compared with other two optimal power distribution methods.

Finally, Chapter 5 discusses a conclusion of this research and discusses the future research for this work.

Chapter 2

Peak Load Forecasting Based On Bayesian Additive Regression Trees

2.1 Introduction

Peak load is defined as the highest aggregated demand of electric power within a period of time in a certain area [83]. To meet the peak load, electrical utilities either schedule the purchase of power or prepare operating reserves. Some, such as peaking plants, are expensive to operate and maintain. Thus, accurate electrical peak load forecasting is critical to power system management to prevent overloading and grid failure [84]. This chapter first focuses on peak load forecasting for a residential area in the New Mexico and further extends to a business area in the Texas. The load data is from the local private electrical utility in the northeast sector of Albuquerque (ABQ), New Mexico, serving approximately 2000 residential and commercial customers where the local utility has a distribution feeder. This locale features a high penetration level of PV generation and is expected to have a significant increase in Plug-in electrical vehicle utilization in near future. The other load data is from the public source of the The Electric Reliability Council of Texas (ERCOT) at the north central of Texas [85]. This business area features a high load factor profile and many PV installations. An accurate prediction of the peak load for the next day is crucial to the electrical

utility. Because according to the prediction, they will make an economical schedule to purchase or generate the power, utilize the renewable energy, maintain or upgrade their electric components in the circuit, and reliably distribute the power to both residential and business areas.

The use of distributed energy resources is increasingly being pursued as a supplement as well as an alternative to large conventional central power stations. Typically, these distributed generators (DG) are renewable energy sources, such as photovoltaic and wind turbines, located throughout the power system. For power load forecasting, the significant implementation of DG as a source of energy in the system adds additional uncertainty because of their dependence on weather. Moreover, the effects on the power-system operation, especially where the intermittent energy source constitutes a significant part of the total system capacity, must be considered.

In the recent literature, [86] proposes a method based on the self-organizing map and support vector machine for short-term utility peak load prediction. In [87], the neural network is trained by a genetic algorithm and used to forecast the peak load based on decomposed data. The semi-parametric additive models is taken in [54] to estimate the relationship between demand and driver variables, such as calendar variables, lagged actual demand and temperature. In general, studies focus on short-term forecasting but few address the predictions of distribution feeder level peak loads, like the residential area in the New Mexico. Load forecast at a distribution feeder level poses challenges because of the complexity of the loads and the possibility of unexpected load operations. Also, in small geographical areas, the distribution of human activities makes predictions difficult whereas when the geographical area is larger, this effect is smoothed out. However, an accurate forecast at a distribution feeder level has many advantages that cannot be reached at a higher level.

The goal of this chapter is to build a peak load prediction model based on a detailed analysis of the load demand considering weather and human factors in the high PV penetration areas. A strong linear relationship is displayed between the peak load and various weather factors during the winter and spring months in Albuquerque and spring, summer and fall in the North Central of Texas. However, a nonlinear relationship dominates in other months. The regression methodology that can accommodate both relationships is considered in this chapter. In particular, the nonparametric regression method, Bayesian Additive Regression Trees (BART), is introduced to

estimate the relationship between the peak load and the weather and human factors. The principled permutation-based inferential approach is utilized in BART to select the most likely effective predictors. In addition, the BART method is compared with other methods such as the Multiple Linear Regression (MLR), Support Vector Machine (SVM) and Gaussian Process Regression (GPR). Each method is tested by forecasting the peak load of the next day. The BART method shows the best predictive result with minimum mean square error (MSE), root mean square error(RMSE), mean absolute forecasting error (MAFE) and R^2. Moreover, the Bayesian probability model within BART provides useful uncertainty estimates which can be used to obtain an accurate Bayesian credible interval for peak load prediction.

This chapter is organized as following: In section 2.2, the load and weather data from the residential and business areas are analyzed. In section 2.3, the BART method is introduced. Section 2.4 constructs the different regression models, which has MLR, SVM, GPR and BART, based on the weather and the human factors and peak load. In section 2.5, a detail results analysis is presented. Finally, conclusion is given in the section 2.6.

2.2 Load And Weather Data Analysis

2.2.1 Load Data Analysis

The models are built using load data obtained from the Supervisory Control And Data Acquisition (SCADA) system in a residential area of Albuquerque, New Mexico in 2012 and 2013. The data was recorded in 15-minute intervals and provides a high fidelity sample. The load data from Texas was obtained from the ERCOT utility public historical load profile in 2014 and 2015 with 15-minute intervals [85]. Note: the original ERCOT load data is measured by kWh in 15-min interval, if you want to get KW load data, you may consider: $ld_{KW} = 4ld_{kWh}$ assuming during that one hour period the $data_{kWh}$ is not changed, where ld means load data. We use the kWh load data to keep the same format as the original load data both in Chapter 2 and Chapter 3. Based on the data, daily load patterns are identified. Typically, the daily Albuquerque load pattern resembles an "M"

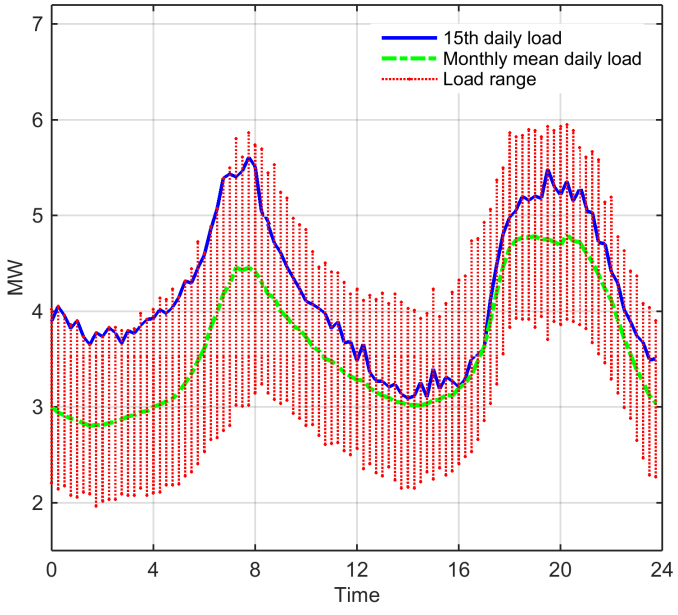

Figure 2.1: January 2013 load shape in Albuquerque

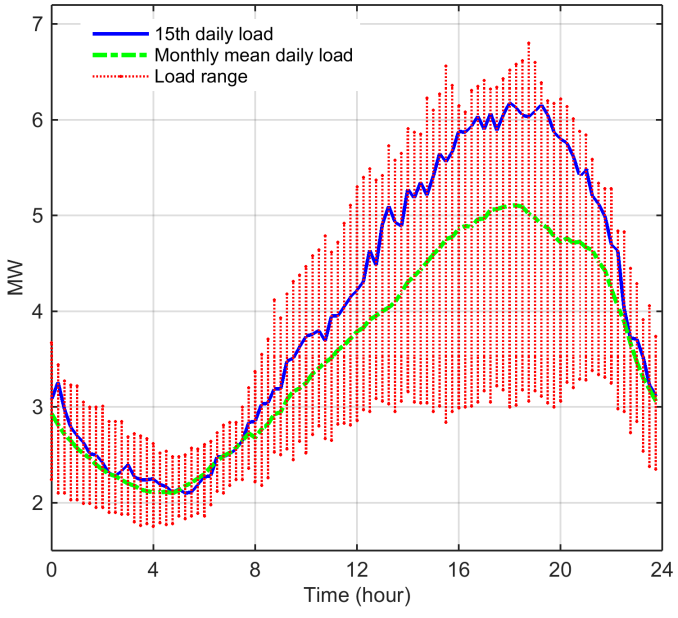

Figure 2.2: July 2012 load shape in Albuquerque

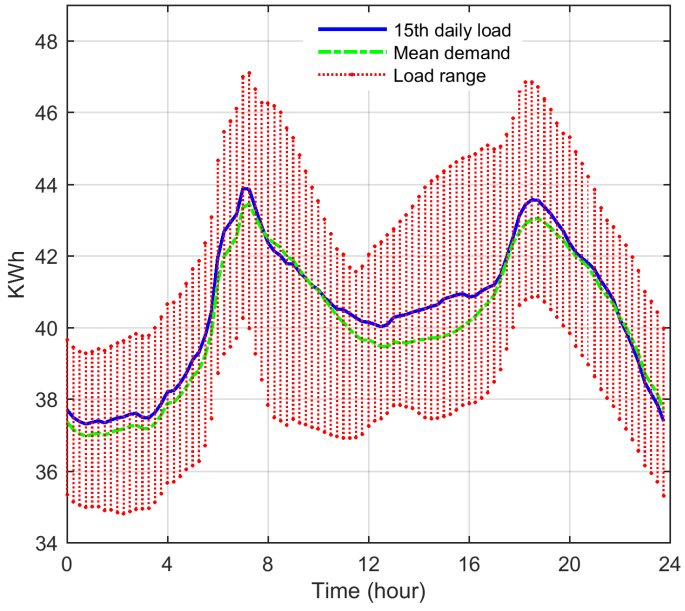

Figure 2.3: Jan 2015 load shape in Texas

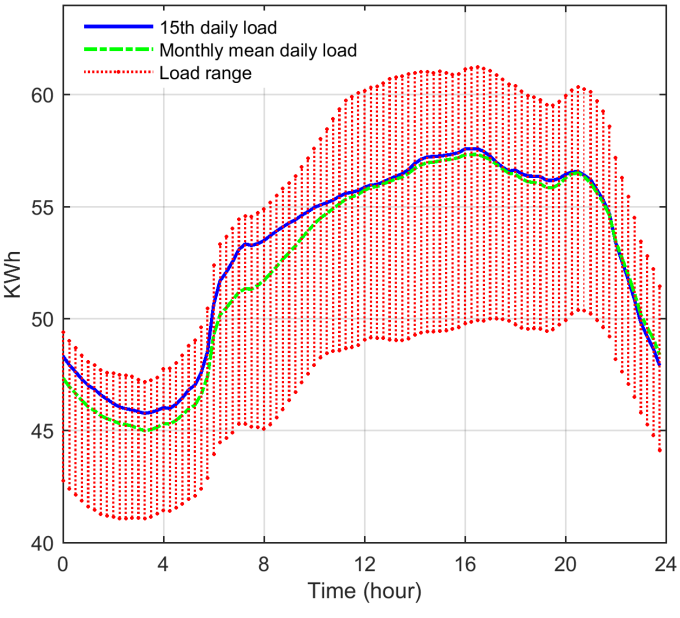

Figure 2.4: July 2014 load shape in Texas

with the peak loads occurring in the morning and evening in the spring, fall and winter months. A similar ”M” shape pattern appears only in the winter and spring in the north central region of Texas. The patterns are shown in Fig. 2.1 and Fig. 2.3. In the summer months in Albuquerque,

the daily load shape is identified as a wave crest with the peak load appearing in the evening and the similar pattern is displayed in the summer and fall of north central of Texas, as shown in Fig. 2.2 and Fig. 2.4. However, there is no specific daily load pattern for the transitional periods from spring to summer and summer to fall in the ABQ and spring to summer in the north central of Texas. Among Figures 1 to 4, the daily load ranges for the month are plotted as red dashed vertical lines in 15-minute intervals for 24 hours. The green dashed line is the monthly mean daily load for each month and the blue solid line displays the daily load on the 15th day of the corresponding month. Both daily load (for example the 15th day of the month) and monthly mean daily load has similar load shapes. The peak load varies with the season. In the spring and fall in Albuquerque, the peak load is usually less than 5 MW while in the summer and winter it is between 6 MW to 7 MW. In north central Texas, the peak load is usually less than 50 KWh during the winter and spring while in the summer and fall it is around 60 KWh. The load demand is low in this area is mainly due to only 23 premises assigned to the high PV load profile. Compared to the residential area in Albuquerque in summer and fall, the peak load in north central Texas appears earlier and lasts longer, which it reflects that is a business area.

2.2.2 Weather Data And Peak Load

We expect the peak load to vary with the weather, so to explore this relationship, weather data from the National Weather Service Forecast Office was obtained for the corresponding periods of the peak load data [88]. For north central Texas, the available weather data includes the daily maximum (tmpmax), mini-mum (tmpmin), average temperature (tmpavg), departure from normal (dep), heating degree day (hdd), cooling degree day (cdd) and precipitation (wtr). In addition to these weather factors, the average wind speed (avespd), two minutes maximum speed (2mxspd), the wind direction (dir), highest wind speed (mxspd) and wind compass direction (dr) are available for Albuquerque weather data. Moreover, to include the effects of human behavior, information on workdays and weekends (wkdt) and holidays (hld) are also added for both areas. These relationships are explored using scatter plots and correlation analysis. The correlation table for the peak load and weather variables are summarized in Table 2.1 for each month in 2012 in Albuquerque

and for north central Texas in 2014 in Table 2.2. Table 2.1 shows that the maximum and average temperature have a high correlation with peak load in nearly every month, and precipitation and heating degree day are highly correlated with peak load in certain months. While in Table 2.2, the maximum and average temperature have a high correlation with peak load in the summer and fall month and hdd, cdd and wkdt are correlated with peak load in the certain month.

Typically, it is not a good choice to either select all the weather factors as the predictors, or take only the weather factors that are highly correlated with peak load as predictors, because they may exist the correlated weather factors, such as daily highest temperature and heating degree day in spring. For linear regression models, forward-stepwise and backward-stepwise may be used in predictors selection [89]. For nonlinear regression models, a method based on a Taylor expansion of the nonlinear model around a given point in the sample space is presented in [90]. To reasonably choose weather factors as predictors for the regression models, different variable selection methodologies are utilized, which is explained in part 2.4.

Table 2.1: Correlation Table Between the Peak Load And Other Factors at 2012 in the Albuquerque

month	tmpmax	tmpmin	tmpavg	dep	hdd	wtr	avespd	2mxspd	dir	mxspd	dr	wkdt	hld
Jan	-0.837	-0.069	-0.591	-0.511	0.591	0.103	0.328	0.238	-0.0745	0.276	0.014	-0.041	0.233
Feb	0.771	-0.1023	-0.645	-0.60	0.644	0.137	0.135	0.196	-0.136	0.175	-0.003	-0.135	0.332
Mar	-0.925	-0.762	-0.909	-0.891	0.909	0.145	0.669	0.531	0.159	0.507	0.155	0	-0.168
Apr	-0.477	-0.305	-0.423	-0.388	0.563	0.79	-0.205	-0.221	-0.011	-0.249	0.105	-0.254	0
May	0.793	0.756	0.838	0.641	-0.387	-0.198	0.01	-0.18	0.118	-0.2	0.179	-0.182	0.004
Jun	0.825	0.633	0.837	0.315	0	0.036	-0.212	-0.309	0.335	-0.277	0.352	0.179	0
Jul	0.795	0.684	0.819	0.815	0	-0.356	-0.088	-0.329	0.126	-0.311	0.095	0.222	0.038
Aug	0.847	0.512	0.845	0.796	0	0.055	0.095	-0.103	0.016	-0.084	-0.085	0.0811	0
Sep	0.867	0.784	0.914	0.705	-0.288	-0.137	-0.167	-0.281	0.408	-0.253	0.443	0.058	0.397
Oct	0.422	0.389	0.444	0.212	-0.351	0	-0.211	-0.009	0.117	-0.018	0.145	-0.05	0
Nov	-0.872	-0.695	-0.859	-0.636	0.859	-0.0163	0.352	0.249	0.007	0.242	0.223	0.087	0.176
Dec	-0.879	-0.72	-0.858	-0.85	0.858	0.056	0.483	0.493	0.404	0.505	0.139	0.01	0.097

2.3 Introduction to the BART Method

Bayesian Additive Regression Trees (BART) is a Bayesian-based regression trees method, introduced by Chipman [91]. BART is the method that is comprised of a sum-of-trees and a set of rules to regulate the prior on the parameters of the method, which not only has the ability to fit the

Table 2.2: Correlation Table Between the Peak Load And Other Factors at 2014 in the north central of Texas

month	tmpmax	tmpmin	tmpavg	dep	hdd	cdd	wtr	wkdt	hld
Jan	-0.693	-0.497	-0.646	-0.655	0.643	0	0.058	-0.512	-0.275
Feb	-0.266	-0.22	-0.262	-0.256	0.268	0	-0.045	-0.513	0.148
Mar	0.551	0.479	0.557	0.533	-0.477	0.766	-0.037	-0.522	0.113
Apr	0.917	0.773	0.893	0.877	-0.819	0.823	-0.368	-0.132	0.045
May	0.804	0.536	0.749	0.795	-0.499	0.735	-0.256	-0.084	-0.232
Jun	0.815	0.565	0.772	0.713	0	0.761	-0.535	-0.416	0.198
Jul	0.860	0.803	0.883	0.86	0	0.879	-0.275	-0.445	-0.196
Aug	0.785	0.742	0.807	0.803	0	0.799	-0.317	-0.459	0.118
Sep	0.919	0.868	0.923	0.811	-0.481	0.92	-0.289	-0.508	0.108
Oct	0.904	0.806	0.93	0.866	-0.371	0.929	-0.089	-0.343	0
Nov	0.194	0.293	0.255	0.171	-0.238	0.242	0.024	-0.413	-0.402
Dec	0.340	0.373	0.391	0.347	-0.387	0.481	-0.038	-0.315	-0.169

interaction, but also has the capability to strongly approximate nonlinearity.

Suppose an unknown function, f, is to be approximated at the ith trial based on an output y_i and let input have a p dimensional space, such that, $\mathbf{x}_i = (x_{i1}, ..., x_{ip})^T$ is a p dimensional vector input at ith trail. The output may be written as:

$$y_i = f(\mathbf{x}_i) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2) \tag{2.1}$$

and define a single tree

$$y_i = g(\mathbf{x}_i; T, M) + \epsilon, \qquad \epsilon \sim N(0, \sigma^2) \tag{2.2}$$

where T is a binary tree that includes a set of terminal nodes (leafs) and decision rules for it's interior nodes. The decision rules split the predictor space based on whether the $\mathbf{x}_i$ belongs to a certain subset of $\mathbf{x}_i$'s range space. The $M = \{\mu_1, \mu_2, ..., \mu_b\}$ consists of a group of parameters, each associated with a terminal node (leaf) of T with all b nodes in T; $g(\mathbf{x}_i; T, M)$ is the function that assigns a value of $\mu_j \in M$ to $\mathbf{x}_i$ based on a given T and M, where $j \in \mathbb{N}, j \in [1, b]$.

With the single tree expression in (2.2), the BART method as a sum-of-trees ensemble can be written as:

$$y_i = \sum_{t=1}^{m} g(\mathbf{x}_i; T_t, M_t) + \epsilon \tag{2.3}$$

where ϵ is a fitting error with $\epsilon \sim N(0, \sigma^2)$; M_t is a set of terminal node parameters that associates with the binary regression tree T_t and $t \in [1, m]$. According to the decision rules in the range of $\mathbf{x}_i$, μ_{jt} from M_t is assigned to the terminal node by function $g(\mathbf{x}_i, T_t, M_t)$. From (2.3), the sum of the terminal values μ_{jt} of all trees is the approximation of y_i. In the BART method, when tree number m bigger than one, every μ_{jt} only consists a small partition of $E(y_i|\mathbf{x}_i)$. However, the μ_{jt} has a large effect on $g(\mathbf{x}_i, T_t, M_t)$ in the case when $g(\mathbf{x}_i, T_t, M_t)$ only depends on a single variable of $\mathbf{x}_i$. Meanwhile, the μ_{jt} has an interactive effect when $g(\mathbf{x}_i, T_t, M_t)$ depends on several variables of $\mathbf{x}_i$. Therefore, when the number of trees is large, the BART method equips the ability of interactive and nonlinear approximation, enhancing its predictive capability.

Compared to other ensemble of trees' methods, BART incorporates a probability strategy. It has the priors and likelihood for the leaf parameters and the tree's structure. The regulation of prior in the BART specializes in the trees' structure, the leaf parameters of the trees, and the error variance σ^2. The prior regulation are simplified in [91] by assuming the independence in the trees' component themselves, the leaf parameters of every tree and the independence of σ^2 with trees' structures and leaf parameters. This can be written as: assume G_t equals $(T_t, M_t), t \in [1, m]$

$$\begin{aligned} p(G_1, ..., G_m, \sigma^2) &= \left[\prod_{t=1}^{m} p(G_t)\right] p(\sigma^2) = \left[\prod_{t=1}^{m} p(T_t, M_t)\right] p(\sigma^2) \\ &= \left[\prod_{t=1}^{m} p(M_t|T_t)p(T_t)\right] p(\sigma^2) = \left[\prod_{t=1}^{m}\prod_{j=1}^{b} p(\mu_{j,t}|T_t)p(T_t)\right] p(\sigma^2) \end{aligned} \tag{2.4}$$

The prior $p(T_t)$ is firstly specified to control the locations of nodes in the trees structure. The nodes have the probability, $\alpha(1+d)^{-\beta}$, to be nonterminal at depth d, where $\alpha \in (0, 1), \beta \in [0, \infty)$. Clearly, when d is big, the probability that a node is nonterminal is small, which forces the tree structure to become shallow. Meanwhile, the splitting variable is randomly selected from predictors with a discrete uniform distribution and the splitting value is selected for the node according to the discrete uniform distribution also.

The prior $p(\mu_{j,t}|T_t)$ is used to control the nodes parameters. Let the $\mu_{j,t} \sim \mathcal{N}(\mu_\mu, \sigma_\mu^2)$. The core

idea to select μ_μ and σ_μ^2 is to make sure that the $E(y_i|\mathbf{x}_i)$ has a high probability of being assigned in the interval $(y_{min,i}, y_{max,i})$, where $y_{min,i}$ and $y_{max,i}$ are the minimum and maximum values at the observation of y in the ith trial. This can be achieved by letting $m\mu_\mu + k\sqrt{m}\sigma_\mu = y_{max,i}$ and $m\mu_\mu - k\sqrt{m}\sigma_\mu = y_{min,i}$. When $k = 2$ for instance, the $E(y_i|\mathbf{x}_i)$ has 95% probability to be assigned in the $(y_{min,i}, y_{max,i})$. Moreover, these steps can be simplified by scaling and shifting the y_i such that $y_{max,i} = 0.5$ and $y_{min,i} = -0.5$. And centralize the prior for $\mu_{j,t}$ at zero so $\mu_\mu = 0$ and let $\sigma_\mu = (2k\sqrt{m})^{-1}$, which yields $\mu_{j,t} \sim \mathcal{N}(0, \sigma_\mu^2)$. It can be seen when the number of tree m or k is big, the σ_μ is forced to small so to shrink the range of $\mu_{j,t}$. Thus, it limits the effect of leafs in the trees.

The last prior $p(\sigma)$ controls the error variance. It is chosen to follow the inverse Chi-squared distribution, $\sigma^2 \sim$ inv-$\chi^2(\nu, \lambda)$. The ν and λ is specified by the data-informed approach, so the estimated $\hat{\sigma}$ of σ would be the y_i standard deviation, or $\hat{\sigma}$ is specified as the residual standard deviation from the model that is built by fitting y_i and $\mathbf{x}_i$ in a least squares linear regression.

In all, the priors provide a regulation of each tree to avoid a domination in the function fitting, thus each tree in BART may be considered a "weak" learner. Moreover, the Markov Chain Monte Carlo (MCMC) algorithm is used to sample the posterior based on the output observation. Furthermore, the BART method does not assume continuity of the response, and thus is appropriate for the case when the response changes suddenly or is nonstationary [92]. Moreover, BART's forecasting ability outperforms other methods, such as random forests, neural networks, boosting and treed Gaussian Process [91] [92]. Because peak loads may change suddenly, BART is used for its prediction. An in-depth description of BART can be found in [91].

2.4 Construction of Models

From chapter 2.2, the strong linear relationship between the peak load and certain weather factors were identified, indicating the multiple linear regression (MLR) method may be appropriate. Next, the support vector machine, (SVM), a kernel based method, is used to fit the peak load and weather

data with its nonlinear approximation capability. In addition, the composite kernels method, Gaussian Process Regression model (GPR), is introduced to further test the kernels method at nonlinear approximation. Last, the BART method is applied to the peak load prediction.

Each of these regression models are tested by out-of-sample testing: let n denote the sample length, such that n daily peak load samples are taken to train each model with n different for each model. Let the $t = 1, \hat{y}_{n+t}$ be the next day peak load forecasting from the trained model and the next day's real peak load be y_{n+t}. The error in the forecasting is defined as $e_t = \hat{y}_{n+t} - y_{n+t}, n \leq 60, t \in [1, 365 - n], \forall t, n \in \mathbb{Z}$. The forecasting error vector $\mathbf{e} = (e_1, e_2, ..., e_t)^T$ is produced by gradually increasing t with a fixed n like a moving window.

2.4.1 The Multiple Linear Regression Method

The general expression of Multiple Linear Regression Method (MLR) is:

$$y_i = \beta_0 + \sum_{k=1}^{p} \beta_k x_{ik} + \epsilon, \quad \epsilon \in N(0, \sigma^2) \tag{2.5}$$

where i is the number of trials; p is the number of predictors; x_{ik}, is the kth predictor at ith trial; y_i is the response; and β_0 and $\beta_k, k \in [1, p]$ are unknown coefficients for the MLR.

In order to perform variable selection and meanwhile, minimize the MLR's forecasting error, the elastic net method is chosen. Elastic net method is the convex combination of L_1 lasso penalty and L_2 ridge penalty, it outweighs the ridge regression in variable selection and surpasses the lasso at dealing with highly correlated predictors and forecasting accuracy [93]. Assume predictors x_{ik} are standardized, the elastic net provides solution for the following parameter estimation [93, 94]:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|^2 + \frac{\lambda}{2}(1 - \alpha)\|\boldsymbol{\beta}\|_{\ell_2}^2 + \lambda\alpha\|\boldsymbol{\beta}\|_{\ell_1} \right] \tag{2.6}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, ..., \beta_p)^T$, with $p = 13$ for the Albuquerque data and $p = 9$ for the north central of Texas data; the predictor $\mathbf{X}$ is a $n \times (p + 1)$ matrix where n is the sample size. And $\mathbf{X} = (\mathrm{I}, \mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p)$ with $\mathbf{x}_k = (x_{1,k}, ..., x_{n,k})^T$ and $k \in [1, p]$ are the weather and human factors and $\mathbf{y}$ is a $n \times 1$ vector of the responses composed of peak load values. The best sample size n is

tested and fixed. When $\alpha = 0$ or $\alpha = 1$, (2.6) corresponds to the case of ridge regression or lasso regression respectively.

2.4.2 Support Vector Machine Model (SVM)

Support Vector Machine (SVM) is a popular and powerful tool for regression analysis. With tests and comparison, the $\epsilon-$Support Vector Regression (ϵ-SVR) is used and the standard form of SVM can be written as [95]:

$$\begin{aligned} \min_{\omega,b,\xi,\xi^*} \quad & \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i + C\sum_{i=1}^{l}\xi_i^* \\ \text{subject to:} \quad & \omega^T\phi(\mathbf{x}_i) + b - y_i \leq \epsilon + \xi_i, \\ & y_i - \omega^T\phi(\mathbf{x}_i) - b \leq \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0, i = 1, ..., l. \end{aligned} \tag{2.7}$$

Where $\mathbf{x}_i = (x_{i1}, ..., x_{ip})^T$, $p = 13$ or $p = 9$ depends on two different areas, is the feature vector, composed of the weather and human factors and i is the sample time index; y_i, $y \in \mathbb{R}^1$, is the target output, that is, the peak load values corresponding to the feature vector; The SVM uses a dual form that depends on dot products: $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ and choosing $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$ and $\epsilon = 0.1$. To perform the feature selection, the method, built on the maximal statistical dependency criterion based on mutual information, is selected [96]. To tune the model, the cross-validation and simulated annealing algorithm are used to select the best pair of parameters: C and γ. The size of the training samples is fixed during the tests.

2.4.3 Gaussian Process Regression Model (GPR)

Gaussian Process (GP), a non-parametric method with a composite of kernels, is used for building the regression model. In this chapter, a brief description of GPR with a composite kernel is

introduced.

A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution [97]. To define a Gaussian Process, the mean function and the covariance function need to be specified. Assume $f(\mathbf{x}_i),\ \forall i \in \mathbb{N}$ is a Gaussian Process, where $\mathbf{x}_i$ is the predictor. It is denoted as: $f(\mathbf{x}_i) \sim \mathcal{GP}(m(\mathbf{x}_i), \mathrm{cov}(\mathbf{x}_i, \mathbf{x}_j)),\ \forall i, j \in \mathbb{N}$ where $m(\cdot)$ is the mean function and $\mathrm{cov}(\cdot, \cdot)$ is the covariance kernel of f.

Let the kernel function is $k(\cdot, \cdot)$ and in this work, the composite kernel function is composed by automatic relevance determination squared exponential (ARDSE) kernel, isotropic weight linear kernel and a constant, which is expressed as:

$$\mathbf{x}_r = \mathbf{x}_k - \mathbf{x}_j, \mathbf{M} = \mathrm{diag}(\ell_1, \ell_2, ..., \ell_p)^{-2}$$
$$k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_f^2 \mathrm{exp}(-\frac{1}{2}\mathbf{x}_r^T \mathbf{M} \mathbf{x}_r) + \sigma_{lr} \mathbf{x}_k^T \mathbf{x}_j + \sigma_c$$

where $\mathbf{M}$ is a symmetric matrix with diagonal values as $\ell_1^{-2}, \ell_2^{-2}, ..., \ell_p^{-2}$. The hyperparameters of the ensemble kernel are: $\boldsymbol{\theta} = (\sigma_f^2, \ell_1, \ell_2, ..., \ell_p, \sigma_{lr}, \sigma_c)$. An ARDSE kernel is the inverse of the hyperparameters of $\mathbf{M}$, which helps the kernel to measure the relevance of new inputs. If it is not relevant, the new inputs will be removed from the inference. In order to fit the linear relation inside predictors and responses, a linear kernel is added and a constant is added to tune the variance. In the simulation, the hyperparameters are initialized as: $\log \ell_t = val,$ where $t \in [1, p]$ and $val \in [5, 7)$. The $\log \sigma_f$ equals the standard deviation of $\mathbf{y}_i$ and $\sigma_{lr} = \sigma_c = 1$. To make the forecasting objective, the val is evenly initialized from $[5, 7)$ with interval 0.2 for every forecasting and the final next day peak load forecasting is the average of all forecasting corresponding to different values of val.

2.4.4 BART Model

In chapter 2.3, the general idea of the nonparametric method, BART, is introduced. Here, for equation (2.3), the $\mathbf{x} = (x_1, x_2, ..., x_p)^T, p = 13$ or $p = 9$, is the weather and human factors and the $\mathbf{y}$ is the vector of peak load value y. For the selection of variables, the principled permutation-based inferential approach is used [98]. The basic idea of this approach is: creating k permutations

of the response vector, $\mathbf{y}_l, l \in [1, k]$, run the BART with each created response and the original predictors, $\mathbf{x}$. Through the permutations, the dependency between the predictors and responses is removed and the likely dependencies among the predictors is saved. Next, summarize the variable inclusion proportion based on the previous k times BART tests. For every predictor, construct a variable inclusion proportion's null distribution. Finally, the methods to choose the threshold are built to select the predictor variables. Cross-validation is used to choose the best thresholding method. It is noticed that the principled permutation-based inferential approach may work better with many predictors. In this work, with few predictors in the data of business area at north central of Texas area, this approach is not applied. To choose the right priors for the BART, our model takes the recommendation from [91] and extends it. The T_t prior is calculated by the backfitting MCMC. The prior of $\mu_{jt}|T_t$ is computed by the distribution of μ_{jt}, where $\mu_{jt} \sim N(0, \sigma_\mu^2), \sigma_\mu^2 = \frac{1}{2k\sqrt{m}}$, m is the number of the tree. Here, choose $k = [1, 1.5, 2, 2.5, 3]$ which evenly distributes among the recommended intervals. For the prior of σ follows the distribution $\sigma^2 \sim \nu\lambda/\chi_\nu^2$ inverse chi-square distribution. The data-informed prior method is used to specify the ν and λ, where $\nu \in [3, 10]$, and a λ is chosen to set the qth quantile of the prior on σ that positioned at its estimation. The pairs of $(\nu, q) = [(3, 0.9), (3, 0.99), (10, 0, 75), (3, 0.95)]$ are chosen. The number of trees are chosen to $m = [10, 30, 50, 100, 150, 200]$. To make the prediction comparatively stable, the burn in and burn out times are set equal 4000 for the MCMC algorithm iteration. The training size of predictors affects the forecasting accuracy and we notice in some case that the default parameters of k, ν, q and m also give good forecasting result. All the simulations are written in R with the *bartMachine* package [99]. The Bayesian based method typically requires heavy computation by nature. In order to run BART, the simulations are tested at the Center for Advanced Research Computing at the University of New Mexico.

2.5 Results

The prediction accuracies for methods: the MLR, the SVM, the GPR and the BART are compared and the effective weather and human factors to forecasting the peak load are summarized for both

Albuquerque and north central of Texas.

2.5.1 Forecasting Comparison

The forecasting accuracy is attained by four measurements of goodness-of-fit: Mean Square Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Forecasting Error (MAFE), coefficient of multiple determination R^2 and the Mean Absolute Percentage Error (MAPE). The four methods are compared in Table 2.3 for the Albuquerque data and in Table 2.4 for the Texas data.

Table 2.3: Forecasting Accuracy Comparison for Albuquerque Area

Index	MLR12	SVM12	GPR12	BART12	MLR13	SVM13	GPR13	BART13
MSE	0.129	0.137	0.125	**0.122**	0.138	0.130	0.125	**0.117**
RMSE	0.360	0.371	0.353	**0.349**	0.371	0.361	0.354	**0.344**
MAFE	0.270	0.276	0.260	**0.258**	0.270	0.270	0.250	**0.248**
R^2	0.863	0.854	0.867	**0.871**	0.826	0.836	0.842	**0.851**
MAPE	5.97%	6.18%	5.79%	**5.73%**	6.03%	6.23%	5.62%	**5.56%**

Table 2.4: Forecasting Accuracy Comparison for north central of Texas

Index	MLR14	SVM14	GPR14	BART14	MLR15	SVM15	GPR15	BART15
MSE	2.010	1.777	1.233	**1.177**	1.872	1.660	1.136	**1.078**
RMSE	1.418	1.333	1.111	**1.085**	1.368	1.288	1.066	**1.038**
MAFE	1.112	0.974	0.837	**0.797**	1.059	0.976	0.788	**0.759**
R^2	0.949	0.955	0.968	**0.970**	0.955	0.960	0.972	**0.974**
MAPE	2.27%	1.98%	1.70%	**1.61%**	2.16%	1.99%	1.62%	**1.53%**

In the above tables, the columns are labeled with the method and the year, so that MLR12 is the 2012 data using the multiple linear regression model. The other columns are similarly denoted. Overall the BART method show the best predictions in all four years and in both the Albuquerque residential area and the Texas business district. It can be noted that the forecasting is superior in the Texas business district because there is less fluctuation in the data. Conversely, the Albuquerque peak load data varies significantly, so the selected predictors in the residential area are hard to make very accurate inference for the response through regression. For example, the fluctuations can be examined by looking at the normalized fluctuations over a year in each district. Using two metrics

of the fluctuation, one using the range and the other the standard deviation, the two districts can be compared. Assume the $\mathbf{y}$ is the yearly peak load record for a certain year at one of these areas. The Data Change Ratio can be define by: $\frac{\max(\mathbf{y})-\min(\mathbf{y})}{\overline{\mathbf{y}}}$, or by Standard Deviation Ratio: $\frac{\sigma_{\mathbf{y}}}{\overline{\mathbf{y}}}$, where $\overline{\mathbf{y}}$ is the mean value of the $\mathbf{y}$.

Table 2.5: Yearly peak Load data changing ratio comparison

	Residential area		Business area	
Measurement Index	2012	2013	2014	2015
Data Change Ratio	0.851	0.840	0.439	0.467
Standard Deviation Ratio	0.202	0.196	0.126	0.129

The data is summarized in Table 2.5. The two measurement indices are consistent and show that the annual fluctuation in Albuquerque is much larger than in North Central of Texas.

A variety of factors can be the source of the large fluctuations in peak power load in the two districts. One explanation could be the difference in usage between the Texas business district and the Albuquerque residential district. However, a more likely source is the effect of weather.

Examination of the peak load data showed Albuquerque had large fluctuations in the spring and summer seasons. In the summer, New Mexico, Arizona and some surrounding areas are subject to the Southwest Monsoonal pattern. Monsoon is an Arabic word which means change in wind pattern. Moist air travels from the Gulf of Mexico and results in abrupt, heavy rainstorms in the summer afternoons which then cools the air temperature significantly. This results in large temperature range even on a daily basis. In the spring, the big temperature diurnal variation is due to the dry, high altitude climate which promotes radiative heat loss during the night but warming to relatively high temperatures during the day. In the Spring, Albuquerques temperature can range over 30 degrees with daily lows in the morning of 58 degrees warming to 88. This is in contrast to the low altitude climate of North Central Texas which shows daily temperature ranges of 19 and 20 degrees in spring and summer respectively.

In order to clarify the prediction errors in the spring and summer, the forecasting prediction error is presented for the four methods for both areas' data for one month in the spring and one

month in the summer. Figure 2.5∼2.8 show that even with the large seasonal temperature swings, the BART method shows more tolerance for fluctuations in the peak load data and yields more accurate forecasts.

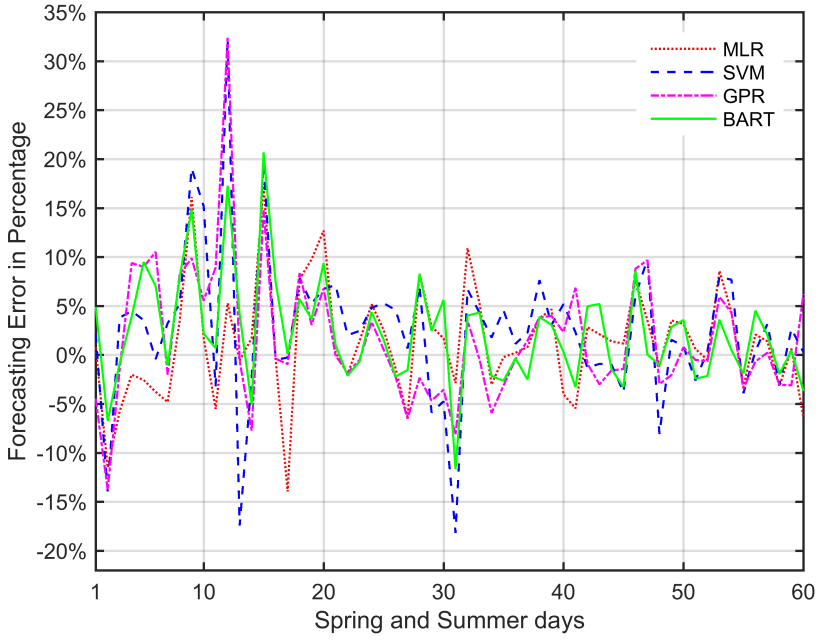

Figure 2.5: 2012 Prediction Error Comparison in Albuquerque

In these figures, the x-axis is the number of the days and y-axis is the forecasting error in percentage, which is defined: $\frac{\hat{y}_d - y_d}{y_d} \times 100\%, \forall d \in [1, 60]$, where the $\hat{y}$ is the peak load forecasting from each method, and the y is the real peak load for the same day. The red dotted line presents the peak load prediction error by using the MLR method, the blue dashed line displays the prediction error for the SVM method, the magenta dash-dot line presents the forecasting error for the GPR model, and the green solid line represents the prediction error from the BART method. From the observation at the figures, the BART forecasting error is distributed closely within ± 0.05 ($\pm 5\%$ error) both in the residential and business areas and has less peak value than the other three types of lines. These demonstrate that BART method has the best generalization and forecasting capability. Another feature of the BART method is the uncertainty estimate, which provides a confidence interval for the prediction. Meanwhile, although MLR is simple, it works well when weather

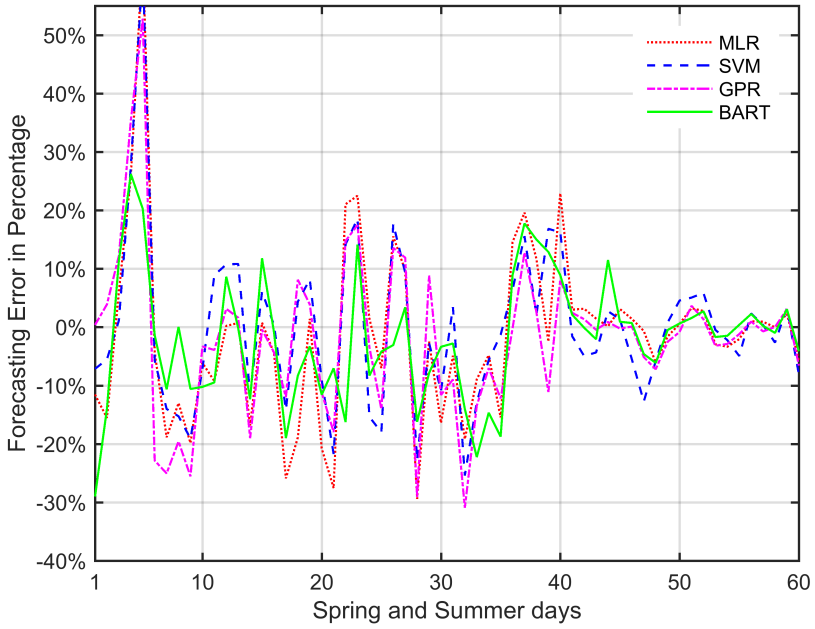

Figure 2.6: 2013 Prediction Error Comparison in Albuquerque

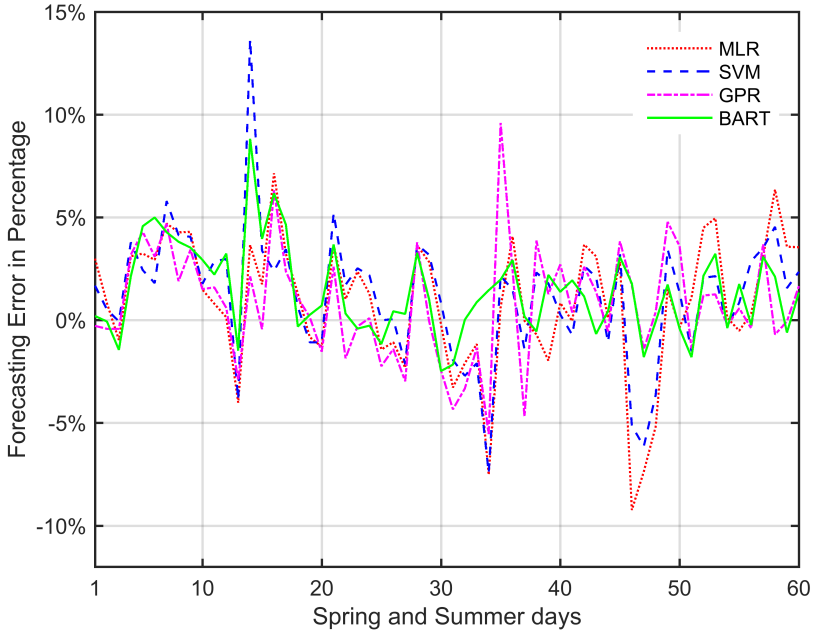

Figure 2.7: 2014 Prediction Error Comparison in north central of Texas

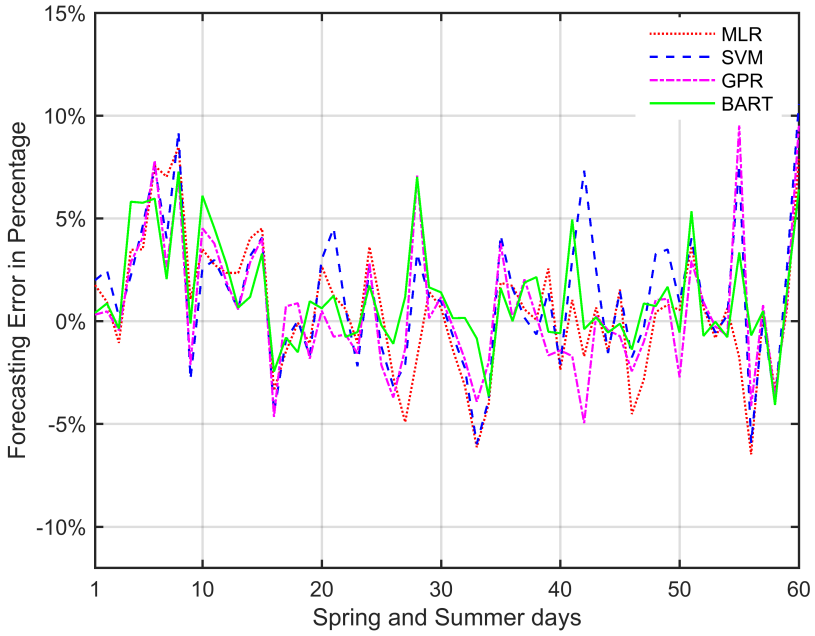

Figure 2.8: 2015 Prediction Error Comparison in north central of Texas

factors are highly correlated with the peak load, which explains why the MLR peak load forecasting model is acceptable to the local utility. However, the BART method exceeds the MLR in every metrics, which provides the local utility with a more accurate peak load prediction. The composite kernel method, GPR, is also good at approximating unknown nonlinear functions. Comparing with the MLR and the SVM, GPR has the value of MSE, RMSE, MAFE, R^2 and MAFE that are closer to BART. GPR requires less computation than the BART, and it also equips the uncertainty estimate. When the computation resource is limited to the user, GPR may be an acceptable choice.

2.5.2 Weather And Human Factor Analysis

For each method, the influence of the dominant weather and human factors affecting the peak load are measured differently. The weather information is kept in the format consistent with [88]. The human factors are attached with two columns as: one column is the work day and weekend which are marked as sequences from 1 to 7 rotatively; the other is the holidays which are marked as 1 and

0 otherwise.

For the MLR, the coefficients for all the predictions selected by elastic net are summarized to measure the importance of weather and human factors by comparing the $\sum_{d=1}^{730} \beta_k$ for each k, where $k \in [1, 13]$ in the residential area at Albuquerque and $k \in [1, 9]$ in the business area at north central Texas. From the ranking, the daily maximum temperature, holidays and the average temperature are the top three influential factors in the residential area. Meanwhile, in addition to previous three influential factors, precipitation is another important weather factor in the business area.

For the SVM, the mutual information quotient scheme of minimum redundancy maximal relevance (mRMR) is performed to select top influential predictor variables [96]. For each step during the forecasting, the selected predictors are used both for training and forecasting. To measure the importance of the predictors, the number of times that each predictor appears during the year long forecasting is counted. The top four influential factors in the SVM model's prediction at residential area are daily maximum and average temperate, temperature departure from normal and the holiday. For the business area, the top four influential factors are the daily maximum and minimum temperature, the temperature departure form normal and the sequence of workdays and weekends.

For the GPR, the kernel hyperparameters, ℓ_p for $p = 9$ or $p = 13$ after tuning, are utilized to measure the importance of the weather and human factors, since the ℓ_p are the *characteristic length-scales*, which measure the relevance between the inputs and the function value [97]. The strategy is to compare the $\sum_{t=1}^{730} \log \ell_p$ for all p in these two areas. The p, which corresponds to the smallest sum of $\log \ell_p$, is the number of the most relevance predictor. By using this strategy, the top four influential factors in the residential area are the daily maximum and average temperature, the sequence of work day and the temperature departure. In the business area, the top four influential factors are the sequence of the work day, the holiday, the daily maximum and average temperature.

Since BART is a nonparametric method, the coefficients cannot be measured directly. The number of times that each factor appears in the variable selection is summarized and compared based on the principled permutation-based inferential method. According to our computations,

the top four influential factors on the peak load are the daily maximum temperature, the daily average temperature, holidays and precipitation in the residential area. In the business area, the daily maximum temperature, the daily average temperature, the sequence of the work day and the cooling degree of day are the top four influential factors.

Therefore, to make an accurate peak load prediction, the daily maximum and average temperature and the human factors are the most important factors for the utility to consider. However, various regression methods favor different weather and human factors, but the strongly correlated ones may always be included inside the predictors.

2.6 Conclusions

In this chapter, Bayesian Additive Regression Trees (BART), a nonparametric method, is introduced for peak load prediction. First, the weather, human factors and load data are analyzed carefully for both residential area at Albuquerque and the business area at north central of Texas. BART is then built based on the principle of permutation-based inferential variable selection. To show the effectiveness of the BART method, it is compared with the MLR, the result that we collaborated on with our local utility, the single kernel method SVM and the composite kernel method GPR. In this work, BART results outperform the other three methods in all four indexes: MSE, RMSE, MAFE and R^2. Furthermore, BART provides a confidence interval to predict the peak load with very high accuracy. The BART method further improves our previous MLR result and can be considered as a superior peak load forecasting model for our local utility.

Chapter 3

Short-term and Very-Short-term Load Forecasting Based On Gaussian Process Regression and Curve Tuning

3.1 Introduction

Short-term load forecasting, as the forecasting length from one hour ahead up to a week [100], and very short-term load forecasting, with forecasting ranging from a few minutes to an hour [118], are the important topics that received a lot of attention in both the research and industrial communities since its forecasting quality is crucial to the operation of smart grid and the trading activities of auction-based electricity market participants [54, 101]. The widely applied integration of distributed generators, such as solar PVs and wind turbines, increase the difficulty of forecasting due to their uncertainty property. The forecasting is becoming even harder for the distribution-level of the circuit, where features small volume of load demand and is easily affected by human and weather factors. This work focuses on the time series day-ahead short-term and very short-term load demand forecasting in the distribution-level circuit at the residential area of Albuquerque of New Mexico and business area at north central of Texas. And both areas have a high penetration

of PV in their load profile.

Many methods have been applied to do time series short-term load forecasting. Classic methods, include the multiple linear regression, stochastic time series, state space, and ARMA, are analyzed in [103, 104]. In [105], the ensemble learning methods are built upon and compared to the ARMA. The development of artificial intelligence influenced several techniques for load forecasting applications. Among them, neural networks have been widely accepted for its strong ability at nonlinear approximation, and two detail reviews are summarized in [21, 38]. Some interesting works, such as [108] introduces a wavelet decomposition based short-term forecasting method and the similar shape functional time series approach is proposed in [109] to do short-term forecasting. In the last two decades, the kernel methods have been studied extensively and employed widely and successfully in the forecasting. In [110], the kernel method, Support Vector Machine (SVM), won the load forecasting competition in European Network on Intelligent Technologies for Smart Adaptive Systems, and an influential review is introduced in [111]. Gaussian Process Regression (GPR) is one of popular kernel methods. It outweighs other kernel methods with probabilistic properties, and it provides the prediction with an uncertainty estimation [97].

In [113, 114], the GPR is introduced to do load forecasting and temperature related load forecasting with common single kernel comparison. In [115], the Twin Gaussian Process priors are applied on both covariance and response for load forecasting. The monthly load is predicted by Gaussian Process with kernels in [116]. Though common kernels are employed and analyzed, the analysis of properties of composite kernels in the GPR framework is barely mentioned. Few hybrid GPR models discuss the composite kernels of GPR. For example, [117] incorporates the neural network model with five GPR sub-models for wind power forecasting and [118] uses the genetic algorithm to optimize the parameters of ensemble GPR sub-models for very short-term load forecasting. However, the concentrations are on the additive models, either linear or nonlinear combination, and do not provide the analysis of product composite kernels.

In this work, a new framework is proposed to do short-term and very short-term forecasting. It combines the composite Matérn kernels (CMKs) (*additive and product of Matérn kernels*) of GPR with curve tuning method, which is built based on the dictionary learning algorithm. First, the

day-ahead level short-term load forecasting model is designed based on the thorough analysis of common kernels in the GPR environment. Both residential and business areas 15-minute interval load data profiles are analyzed in detail. The amplitude of load patterns is usually different between weekdays and weekends in the business area, while it has less difference in the residential area, but the daily load pattern changes more in the residential area. The common kernels are tested with different length of training data for four years to select the best kernel. Among them, the Matérn kernel displays the best approximation properties and the most robust load forecasting corresponding to the different parameter initialization. The CMKs are then designed based on the Matérn kernel. To further improve the composite kernels forecasting accuracy, an algorithm is designed to select the training data from historical load data based on the multivariate normal distribution of daily maximum and minimum temperatures. While CMKs with selected training data can provide a comparatively high accuracy and uncertainty estimation of day-ahead load forecasting, there are some cases that the CMKs can't deal, such as weather is suddenly changed next day or human activities are not expected as usual. Even in these cases, in some sense, the load patterns share some similarities with the normal case. To solve the issues in these cases and to take advantage of the similarities, a new error drove curve tuning algorithm based on the dictionary learning algorithm, K-SVD [120], is proposed to adjust the predicted curve, derived from the CMKs. When the accumulated error, generated between predicted curve and real load curve, surpasses the defined boundary, the algorithm first decomposes the predicted curve into several atoms based on the fixed dictionary. The partial known load curve is then approximated by the same partial length of some atoms generated from sparse coding. The same coefficients and atoms with whole day length are then employed to do curve tuning and forecasting. The minimization of the variance of coefficients is the first index that is utilized to select the best curve from sparse coding for the specific number of atoms. To further stabilize and converge the curve tuning, the mean value of the predicted load curve is an additional index that is used to measure the quality of the predicted curves. Since the mean value of the predicted load curve is an inference for the mean value of real load curve, the model to adjust the inference is designed by integrating the linear regression models with forecasting errors as feedback.

The main contributions of this work are:

(1) A new framework is proposed for short-term and very short-term load forecasting with high accuracy for both residential and business areas with high PV penetration at the feeder level.

(2) The CMKs based on GPR are proposed to do time series short-term load forecasting. The detailed kernels comparison and Matérn kernels are analyzed.

(3) An algorithm based on the daily maximum and minimum temperature is proposed to help the CMKs to choose the training data, which improves the day-ahead forecasting accuracy to a considerable. The training data with weekend and weekday is separated also improves the forecasting accuracy.

(4) A novel curve tuning algorithm based on the dictionary learning algorithm, K-SVD, is proposed. This curve tuning algorithm displays excellent ability at very short-term load forecasting (from 15-minutes to 4 hours). In addition, it takes the advantage of CMKs' outputs, and it continuously improves the kernels' output with time.

(5) The curve tuning algorithm is adjusted to converge quickly by integrating linear regression models with forecasting error as feedback. The method is designed to infer the mean value of the daily load.

3.2 Introduction to GPR and K-SVD

The GPR is briefly reviewed and the dictionary learning algorithm, K-SVD with order recursive matching pursuit, is also briefly introduced.

3.2.1 Gaussian Process Regression (GPR)

Gaussian Process (GP), a non-parametric method with a composite of kernels, is used for building regression model. GP provides a principled and probabilistic approach to learning in kernel

machines, which provide the GP with advantages at model selection and interpretation of model predictions [97].

Let $\hat{y}_i$ be the prediction of the load y_i where i is the sample daily time index and $i \in \mathbb{N}$. Define the prediction:

$$\hat{y}_i = f(\mathbf{x}_i) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}_i) = \sum_{m=1}^{h} w_m \phi_m(\mathbf{x}_i) \tag{3.1}$$

where $\boldsymbol{\phi}(\cdot)$ is the vector of h basis function $\phi(\cdot)$. The predictor, $\mathbf{x}_i$, is the vector with length p. For the time sequence, $p = 1$ and $\mathbf{x}_i$ is the time variable. $\mathbf{w}$ is the weight vector and $\mathbf{w} = (w_1, w_2, ..., w_h)^T$ and assume the Gaussian distribution of $\mathbf{w}$ that is defined: $P(\mathbf{w}) = \mathcal{N}(0, \sigma_w^2 \mathbf{I})$, where $P(\mathbf{w})$ is the prior distribution, σ_w^2 is the variance and $\mathbf{I}$ is the unit matrix [112].

Let $\hat{\mathbf{y}}_i = (\hat{y}_{i-l+1}, \hat{y}_{i-l+2}, ..., \hat{y}_i)^T$ be the set of responses to the certain predictors $\mathbf{x}_{i-l+1}, \mathbf{x}_{i-l+2}$, ..., $\mathbf{x}_i$, where l is the training length with $l < i, l \in \mathbb{N}$. Then $\hat{\mathbf{y}}_i$ may be written as:

$$\hat{\mathbf{y}}_i = \Phi \mathbf{w} \tag{3.2}$$

where Φ is matrix with element $\Phi_{uj} = \phi_u(\mathbf{x}_j), \forall u \in [1, 2, ..., h], \forall j \in [i - l + 1, i - l + 2, ..., i]$. It can be seen that $\hat{\mathbf{y}}_i$ follows a Gaussian distribution because it is a linear combination of Gaussian variables, and thus, the mean and covariance of $\hat{\mathbf{y}}_i$ may be written as [42]:

$$\mathbb{E}(\hat{\mathbf{y}}_i) = \Phi \mathbb{E}(\mathbf{w}) = \mathbf{0} \tag{3.3}$$

$$\mathrm{cov}(\hat{\mathbf{y}}_i) = \mathbb{E}(\hat{\mathbf{y}}_i \hat{\mathbf{y}}_i^T) = \Phi \mathbb{E}(\mathbf{w}\mathbf{w}^T)\Phi^T = \sigma_w^2 \Phi \Phi^T \tag{3.4}$$

To build a GP regression model, assume

$$y_i = \hat{y}_i + \epsilon_i \tag{3.5}$$

where ϵ_i is the Gaussian noise as the prediction difference and $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Let $\mathbf{y}_i = (y_{i-l+1}, y_{i-l+2}, ..., y_i)^T$, then, by combining (3.5) and definition of $\hat{\mathbf{y}}_i$, it can be deduced:

$$P(\mathbf{y}_i) = \mathcal{N}(\mathbf{0}, \mathrm{cov}(\hat{\mathbf{y}}_i) + \sigma_\epsilon^2 \mathbf{I}) \tag{3.6}$$

This indicates that the prior distribution of $\mathbf{y}_i$ is also Gaussian distribution [113] and the covariance of $\mathbf{y}_i$ is $\mathrm{cov}(\mathbf{y}_i) = \mathrm{cov}(\hat{\mathbf{y}}_i) + \sigma_\epsilon^2 \mathbf{I}$

Let (k, j) be the entry of $\mathrm{cov}(\hat{\mathbf{y}}_i)$. The element of $\mathrm{cov}(\hat{\mathbf{y}}_i)$ may be written as:

$$\mathrm{cov}(\hat{\mathbf{y}}_i)_{kj} = [\sigma_w^2 \Phi \Phi^T]_{kj} = \sigma_w^2 \sum_{m=1}^{h} \phi_m(\mathbf{x}_k)\phi_m(\mathbf{x}_j) \tag{3.7}$$

The element of $\mathrm{cov}(\mathbf{y})$ thus may be written as:

$$\mathrm{cov}(\mathbf{y}_i)_{kj} = \sigma_w^2 \sum_{m=1}^{h} \phi_m(\mathbf{x}_k)\phi_m(\mathbf{x}_j) + \sigma_\epsilon^2 \delta_{kj} \tag{3.8}$$

where δ_{kj} is the Dirac function.

The calculation of the inner product of (3.7) and (3.8) can be computed as the covariance functions, which satisfy the Mercer conditions [97], to avoid the explicit computation of $\phi(\cdot)$. The covariance function (kernel) may be written as:

$$k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_w^2 \sum_{m=1}^{h} \phi_m(\mathbf{x}_k)\phi_m(\mathbf{x}_j) \tag{3.9}$$

To make a prediction of y_{i+1} conditioned on $\mathbf{y}_i$, $P(y_{i+1}|\mathbf{y}_i)$, the posterior distribution is computed by Bayesian theorem [113]:

$$P(y_{i+1}|\mathbf{y}_i) = \frac{P(y_{i+1}, \mathbf{y}_i)}{P(\mathbf{y}_i)} \tag{3.10}$$

where $P(y_{i+1}, \mathbf{y}_i)$ is the joint distribution that follows a Gaussian distribution by previous analysis and (3.6). $P(\mathbf{y}_i)$ is the prior distribution which is a constant.

The joint distribution may be expressed as:

$$P(y_{i+1}, \mathbf{y}_i) = \mathcal{N}(0, \mathrm{cov}(\mathbf{y}_{i+1})) \tag{3.11}$$

where $\mathrm{cov}(\mathbf{y}_{i+1})$ is the covariance matrix with dimension $(l+1) \times (l+1)$. It can be expressed as:

$$\mathrm{cov}(\mathbf{y}_{i+1}) = \begin{pmatrix} \mathrm{cov}(\mathbf{y}_i) & \mathbf{k}_v \\ \mathbf{k}_v^T & s \end{pmatrix} \tag{3.12}$$

where $\mathrm{cov}(\mathbf{y}_i)$ is a matrix with $l \times l$ dimension. $\mathbf{k}_v$ is $l \times 1$ vector that has elements $k(\mathbf{x}_n, \mathbf{x}_{i+1})$ for $n = i - l + 1, i - l + 2, ..., i$ and s is a scalar [42].

With (3.10)$\sim$(3.12), matrix partition and inversion, the mean and covariance of the conditional distribution $P(y_{i+1}|\mathbf{y}_i)$ are expressed below, which are the results for the GP regression prediction:

$$m(y_{i+1}) = \mathbf{k}_v^T \mathrm{cov}^{-1}(\mathbf{y}_i)\mathbf{y}_i \tag{3.13}$$

$$\sigma^2(y_{i+1}) = s - \mathbf{k}_v^T \mathrm{cov}^{-1}(\mathbf{y}_i)\mathbf{k}_v \tag{3.14}$$

To make a good prediction in GP regression model, it not only needs to choose good kernel/kernels, but also requires finely inferring and tuning the hyperparameters. The detailed comparison of common kernels and the general idea for inferring the hyperparameters are introduced in Section 3.3. Note, since the data sampled here is in 15-min intervals for day-ahead forecasting, another 95 time points, i.e.: $y_{i+2}, y_{i+3}, ..., y_{i+96}$, are predicted in a similar way.

3.2.2 Dictionary Learning Algorithm (K-SVD)

A dictionary is defined as a collection of atoms (vectors) that can be used for signal representation. The representation is a linear combination of some atoms from the dictionary to exactly or approximately express the original signal [119]. Let $\mathbf{y} \in \mathbb{R}^n$ be a target signal, the $\mathbf{D} \in \mathbb{R}^{n \times m}$ be a overcomplete dictionary that has m columns as the atoms $\{\mathbf{d}_i\}_{i=1}^m$. The dictionary learning tries to find the coefficients $\mathbf{k}, \mathbf{k} \in \mathbb{R}^m$ that satisfies the exact representation $\mathbf{y} = \mathbf{Dk}$, or approximate representation $\mathbf{y} \approx \mathbf{Dk}$ with $||\mathbf{y} - \mathbf{Dk}||_p < \epsilon$, for a small value of ϵ and some L^p norm. For the K-SVD, the L^2 norm is used.

The K-SVD is one of the most popular dictionary learning algorithms for signal sparse representation. The algorithm generalizes the K-means clustering process and efficiently solves the following problem:

$$\min_{\mathbf{D},\mathbf{K}} \left\{ ||\mathbf{Y} - \mathbf{DK}||_F^2 \right\}, \quad \text{subject to} \quad \forall j, ||\mathbf{k}_j||_0 \leq T_0 \tag{3.15}$$

where $\mathbf{Y}$ is the matrix of column signals $\{\mathbf{y}_i\}_{i=1}^m$, $\mathbf{K}$ is the matrix of coefficients $\{\mathbf{k}_j\}_{j=1}^m$, the letter F symbolizes the Frobenius norm, and $||\cdot||_0$ is the L^0, which counting the nonzero entries of $\mathbf{k}_j$ should less or equal to the predefined nonzero natural number T_0 [120].

The K-SVD algorithm solves (3.15) mainly by alternating between two steps: one is the sparse coding based on the fixed dictionary $\mathbf{D}$, and the other is updating the dictionary with fixed coefficients $\mathbf{K}$.

In the sparse coding step, the K-SVD is trying to find the best coefficients $\mathbf{K}$ when fixed library $\mathbf{D}$. At this step, the penalty term of (3.15) can be decoupled into m individual problems that have the expression:

$$\min_{\text{fixed }\mathbf{D},\mathbf{k}_j} \left\{||\mathbf{y}_j - \mathbf{D}\mathbf{k}_j||_2^2\right\}, \quad \text{subject to } ||\mathbf{k}_j||_0 \leq T_0, \forall j \in [1, m] \tag{3.16}$$

(3.16) is finely solved by the pursuit algorithms, but when T_0 is too small, the optimal $\mathbf{k}_j$ may not exist. In this work, the order recursive matching pursuit (ORMP) algorithm is used.

In the dictionary updating step, the coefficient matrix $\mathbf{K}$ is fixed after the sparse coding. It is unlikely to update whole dictionary $\mathbf{D}$ at once, so the algorithm updates one column of $\mathbf{D}$ each time while fixing the other columns. For example, take the case when only the tth column $\mathbf{d}_t$ of dictionary $\mathbf{D}$ is updated, the penalty item of (3.15) can be written as:

$$\begin{aligned} ||\mathbf{Y} - \mathbf{D}\mathbf{K}||_F^2 &= \left|\left|\mathbf{Y} - \sum_{i=1}^m \mathbf{d}_i\mathbf{k}_J^i\right|\right|_F^2 = \left|\left|\left(\mathbf{Y} - \sum_{i\neq t} \mathbf{d}_i\mathbf{k}_J^i\right) - \mathbf{d}_t\mathbf{k}_J^t\right|\right|_F^2 \\ &= ||\mathbf{F}_t - \mathbf{d}_t\mathbf{k}_J^t||_F^2 \end{aligned} \tag{3.17}$$

where $\mathbf{k}_J^t$ is the tth row of coefficient matrix $\mathbf{K}$ and the $\mathbf{F}_t$ symbolizes the error between the signals and the fixed part of dictionary without the tth column.

When using the singular value decomposition (SVD) to solve the equation (3.17), a restriction strategy, removing the zero entries in $\mathbf{F}_t$ and $\mathbf{k}_J^t$, is applied to avoid the $\mathbf{k}_J^t$ to be filled. The principle idea is to define the set of nonzero entry indices of $\mathbf{k}_J^t$, such that $\omega_t = \{i|1 \leq i \leq m, \mathbf{k}_J^t(i) \neq 0\}$, which pointing to $\{\mathbf{y}_i\}$ that use the atom $\mathbf{d}_t$. Define the matrix $\mathbf{\Omega}_t$ such that it has ones on the

$(\omega_t(i), i)$th entries and zeros otherwise and $\mathbf{\Omega}_t$ is $m \times |\omega_t|$ size. Then update the $\mathbf{F}_t$ and $\mathbf{k}_J^t$ in (3.17):

$$||\mathbf{F}_t\mathbf{\Omega}_t - \mathbf{d}_t\mathbf{k}_J^t\mathbf{\Omega}_t||_F^2 = ||\mathbf{F}_t^r - \mathbf{d}_t\mathbf{k}_R^t||_F^2 \tag{3.18}$$

Then, (3.18) can be solved directly by SVD as $\mathbf{F}_t^r = \mathbf{U}\mathbf{\Delta}\mathbf{V}^T$, with $\mathbf{d}_t$ equals the first column of $\mathbf{U}$ and $\mathbf{k}_R^t$ equals the first column of $\mathbf{V} \times \mathbf{\Delta}(1,1)$. After updating the dictionary, the atoms in the dictionary $\mathbf{D}$ are normalized due to the SVD, and the support of all representations is improved when $k_s \neq 0, \forall k_s \in \mathbf{k}_R^t$ or stays the same when $k_s = 0$ [120]. The next step turns to the sparse coding, and then updating the dictionary $\mathbf{D}$ iteratively.

The K-SVD algorithm is widely used in the image and audio processing but has not been used in regression applications. In this work, the K-SVD is used for two purposes: one is in decomposing the day-ahead load forecasting results from GPR kernel into a new overcomplete dictionary. The other is generating the coefficients to tune the atoms based on the known daily load.

3.3 Kernels And Day-ahead Forecasting

In this section, the common GPR kernels are analyzed and compared, and applied to residential and business area data for short-term load forecasting. The algorithm of data selection for training the kernel is also proposed.

3.3.1 Kernel Selection and Properties Analysis

The kernel (covariance function) is the crucial ingredient in a Gaussian process predictor, as it encodes the assumptions about the function which people wish to learn [97]. There are many kernels that have been proposed, the squared exponential (SE) kernel $\big(k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_f^2 \exp(-\frac{1}{2\ell^2}\mathbf{x}_r^T\mathbf{x}_r)\big)$, the rational quadratic (RQ) kernel $\big(k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_f^2(1 + \frac{1}{2\alpha\ell^2}\mathbf{x}_r^T\mathbf{x}_r)^{-\alpha}\big)$, the periodic kernel (PERD) $\big(k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_f^2 \exp(-\frac{2}{\ell^2}\sin^2(\pi||\mathbf{x}_r||/p))\big)$, the Matérn kernels $\big([\text{MATRN1}] : \nu = 0.5, k(\mathbf{x}_k, \mathbf{x}_j) =$

$\sigma_f^2 \exp(-\frac{1}{\ell}\sqrt{\mathbf{x}_r^T\mathbf{x}_r})$; [MATRN2] : $\nu = 1.5, k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_f^2(1 + \frac{\sqrt{3\mathbf{x}_r^T\mathbf{x}_r}}{\ell})\exp(-\frac{1}{\ell}\sqrt{3\mathbf{x}_r^T\mathbf{x}_r})\big)$, the neural network (NN) kernel $\big(k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_f^2 \sin^{-1}(\frac{\mathbf{x}_k^T \sum \mathbf{x}_j}{\sqrt{\ell^2 + \mathbf{x}_k^T \sum \mathbf{x}_k}\sqrt{\ell^2 + \mathbf{x}_j^T \sum \mathbf{x}_j}})\big)$, where $\sum$ is the weight matrix in the neural network kernel, the compact support, piecewise polynomial (POLY) $\big(k(\mathbf{x}_k, \mathbf{x}_j) = \sigma_f^2 \max(0, 1-d)^{j+g} f_g(d)\big)$, where $d = \frac{||\mathbf{x}_r||}{\ell}, j = |\frac{D}{2}| + g + 1, D = 1, g = 3$ and when $g = 3, f_g(d) = (1-d)^{j+3} + \frac{(j^3+9j^2+23j+15)d^3 + (6j^2+36j+45)d^2 + (15j+45)d + 15}{15}$and the Gabor $\big(k(\mathbf{x}_k, \mathbf{x}_j) = \exp(-\frac{\mathbf{x}_r^T\mathbf{x}_r}{2\ell^2}\cos(2\pi\mathbf{x}_r/p)\big)$ are the commonly-used kernels among them. To approximate the different features of data, there may exist one or more kernels to fit the unknown function and make good forecasting of the future data. To select the best kernel as the ingredient to build a composite kernel, which does time series day-ahead forecasting, the commonly-used kernels are compared in this work with different length of training data.

In practice, besides selecting the covariance function (kernel), inferring and tuning the parameters of the functions from the data play another important role for forecasting in the GPR [42]. The shared hyperparameters among previous kernels are represented by $\boldsymbol{\theta} = (\sigma_f, \ell)$, where the σ_f controls the standard deviation of the response and the ℓ controls the length scale of the correlations. To learn and infer the hyperparameters $\boldsymbol{\theta}$ from the data, the common approach is utilized by maximizing the log marginal likelihood function $p(\mathbf{y}_i|\boldsymbol{\theta})$ to estimate the $\boldsymbol{\theta}$. The log likelihood function and the partial derivatives of the likelihood with respect to the hyperparameters may be written as [97]:

$$\log p(\mathbf{y}_i|\boldsymbol{\theta}) = -\frac{1}{2}\log|\mathrm{cov}_{\mathbf{y}_i}| - \frac{1}{2}\mathbf{y}_i^T \mathrm{cov}_{\mathbf{y}_i}^{-1}\mathbf{y}_i - \frac{1}{2}\log(2\pi) \tag{3.19}$$

$$\frac{\partial}{\partial\theta_t}\log p(\mathbf{y}_i|\boldsymbol{\theta}) = -\frac{1}{2}\mathrm{tr}\left(\mathrm{cov}_{\mathbf{y}_i}^{-1}\frac{\partial \mathrm{cov}_{\mathbf{y}_i}}{\partial\theta_t}\right) + \frac{1}{2}\mathbf{y}_i^T \mathrm{cov}_{\mathbf{y}_i}^{-1}\frac{\partial \mathrm{cov}_{\mathbf{y}_i}}{\partial\theta_t}\mathrm{cov}_{\mathbf{y}_i}^{-1}\mathbf{y}_i \tag{3.20}$$

where $\mathrm{cov}_{\mathbf{y}_i} = \mathrm{cov}(\mathbf{y}_i)$ and $\mathrm{tr}(\cdot)$ is the trace of the matrix. Note, since $\log p(\mathbf{y}_i|\boldsymbol{\theta})$, in general, is a nonconvex function, the marginal likelihood may have multiple local maxima [97].

The product of Matérn kernels (PMK), which is expressed in (3.21) (MTN2*MTN2), is pro-

posed for the residential area since it is slightly better than the MTN2:

$$\begin{aligned} \mathbf{x}_r &= \mathbf{x}_k - \mathbf{x}_j, r_i = \frac{1}{\ell_i}\sqrt{\mathbf{x}_r^T \mathbf{x}_r}, \forall i \in \{1,2\} \\ k(\mathbf{x}_k, \mathbf{x}_j) &= \sigma_{f_1}^2 \sigma_{f_2}^2 \frac{(1+\sqrt{3}r_1)^2(1+\sqrt{3}r_2)^2}{\exp(\sqrt{3}(r_1+r_2))} \end{aligned} \tag{3.21}$$

The additive of Matérn kernels (AMK) is employed to learn the business area data (MTN1 + MTN1), and it is expressed by:

$$\begin{aligned} r_n &= \frac{1}{\ell_n}\sqrt{\mathbf{x}_r^T \mathbf{x}_r}, \forall n \in 1,2 \\ k(\mathbf{x}_k, \mathbf{x}_j) &= \sigma_{f_1}^2 \exp(-r_1) + \sigma_{f_2}^2 \exp(-r_2) \end{aligned} \tag{3.22}$$

3.3.2 Data Features of Residential and Business Areas

Two years (2012 and 2013) of private load data for the residential area is taken from the local utility at the Albuquerque and two years (2014 and 2015) of public data for the business area is taken from the ERCOT at the north central of Texas [85]. The features varied due to the different number and activities of people, and the weather in these areas. Four figures, 2012 and 2013 from the Albuquerque, and 2014 and 2015 from the Texas, are plotted below for explanation.

From the figures 3.1 to 3.4, the upper subplot is the load curve for the whole year; the middle subplot is the load curve for a month, whose load curve is considered to be comparatively smoother than other months in a half-year corresponding to the upper subplot; Note: in order to see the load curve pattern clearly in a week, the load curve smoother month is chosen. The lower subplot is the load curve for the first whole week of the corresponding month from the middle subplot. Take the figure 3.1 for example, the middle subplot plots the load curves in the April corresponding to the upper subplot, where the whole 2012 year load curves are plotted, because the load curves in April are considered to be smoother than other months (Jan, Feb, Mar, May and Jun) in the first half year of 2012. While the lower subplot plots the first whole week of the April from the April 2nd to the April 9th. The figure 3.2 follows the same pattern as 3.1, but the middle subplot plots the load curves in the October of 2013, because the curves in October is considered to be smoother

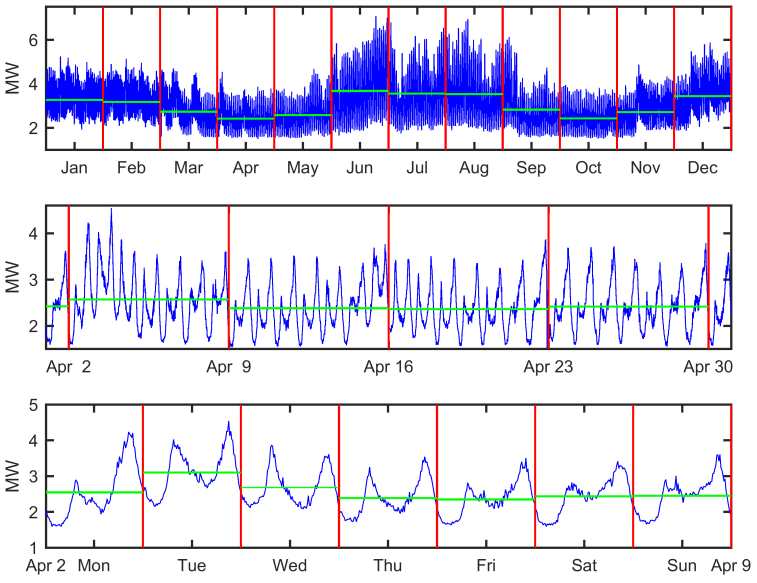

Figure 3.1: 2012 data from residential area at Albuquerque (Yearly load plotted in the upper subplot; April load plotted in the middle subplot; the load at first week of April plotted in the lower subplot)

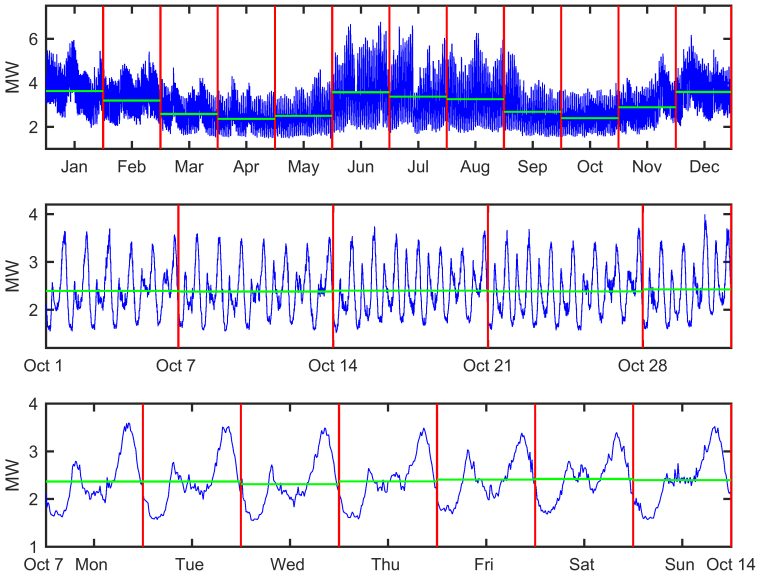

Figure 3.2: 2013 data from residential area at Albuquerque with (Yearly load plotted in the upper subplot; October load plotted in the middle subplot; the load at first week of October plotted in the lower subplot)

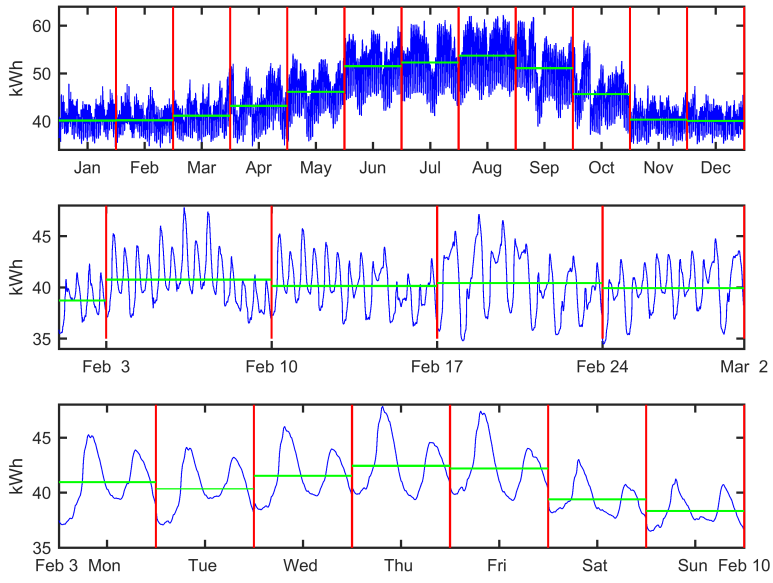

Figure 3.3: 2014 data from business area at Texas (Yearly load plotted in the upper subplot; February load plotted in the middle subplot; the load at first week of February plotted in the lower subplot)

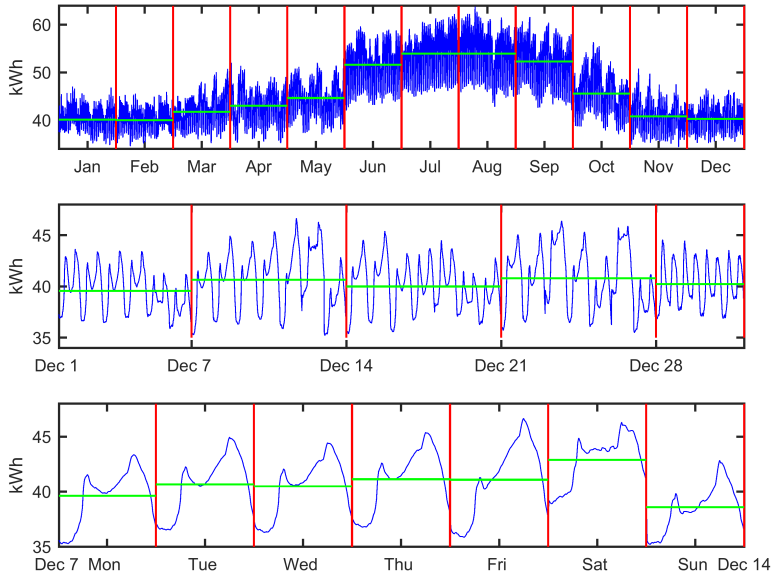

Figure 3.4: 2015 data from business area at Texas (Yearly load plotted in the upper subplot; December load plotted in the middle subplot; the load at first week of December plotted in the lower subplot)

than any months (Jul, Aug, Sep, Nov, Dec) in the second half year of 2013. The same area, the yearly data has some common features, showing different months in the first and second half year for the different year would display more information in this area than just plotting the load curve in the same month at each year. The same ideology is applied to the figures 3.3 and 3.4, where the load curves in the February is considered to be smoother in the first half year of 2014. The load curves in December is smoother in the second half year of 2015. Meanwhile, in these figures, the horizontal green lines are the mean load value during the time period inside of the two vertical red lines.

The residential area has a larger volume of load demand, which is megawatt level than the kilowatt level in the business area. The daily pattern is illustrated in the load curves in the smoother month for each year plotted in the middle subplot, where mean load values for each week inside of the month have fewer differences. Comparing the lower subplots in these figures, the business area has clear different load patterns between the weekday and weekend. In addition, from the upper subplot, as the green lines displayed, the business area has the bigger mean load value variance between every month. These differences between the two areas display that the load demand in the business area fluctuates wider in numbers. However, the load curve in the residential area fluctuates more by the monthly mean load demand ratio (MMLDR), $\frac{1}{11}\frac{\sum_{mth=1}^{11}\left|\text{MMLD}_{mth}-\text{MMLD}_{mth+1}\right|}{\text{yearlyMeanLoad}}$. The MMLDR at 2013 is 0.146 and at 2015 is 0.055. These imply the prediction may harder at the residential area, and the weekend and weekday data may need to be treated separately. The presumable reasons for these differences: first, people in the residential area may somehow have similar power consumption pattern during the weekday or weekend, but the load curve differences is lagrer by mean load ratio. The power consumption pattern is clearly different in business area between the weekday and weekend may due to the activities and the business property itself. Second, the residential area has about 2,000 settled people while the business area has floating population, which introduces the difference in their power consumption grades.

To compare the kernels and infer the hyperparameters for the time series day-ahead load forecasting, data for four years is trained and tested. The results are summarized in the following four tables (tables 3.1 to 3.4) respectively. With $\mathbf{x}$ is the time interval ranging form 1 to 96 and $\mathbf{y}$ is

Table 3.1: forecasting MSE for kernels with different length of training data at 2012

	intialization $\ln\ell = 0.1 : 0.1 : 1$					intialization $\ln\ell = 0.1 : 0.1 : 3.2$				
Kernel	1day	2days	3days	4days	5days	1day	2days	3days	4days	5days
SE	0.130	0.134	0.143	0.149	0.154	0.183	0.175	0.176	0.175	0.176
RQ	0.128	0.133	0.142	0.148	0.153	0.153	0.147	0.151	0.153	0.156
PERD	0.295	0.307	0.353	0.357	0.350	0.204	0.205	0.221	0.222	0.220
NN	0.132	0.133	0.143	0.148	0.153	0.179	0.162	0.158	0.158	0.161
POLY	0.128	0.133	0.143	0.148	0.153	0.138	0.137	0.144	0.149	0.156
Gabor	0.129	0.134	0.144	0.150	0.139	0.164	0.155	0.161	0.157	0.150
MATRN1	0.132	0.133	0.142	0.147	0.152	0.132	0.133	0.142	0.147	0.152
MATRN2	**0.128**	**0.132**	**0.142**	**0.147**	**0.152**	**0.128**	**0.132**	**0.142**	**0.147**	**0.152**

Table 3.2: forecasting MSE for kernels with different length of training data at 2013

	intialization $\ln\ell = 0.1 : 0.1 : 1$					intialization $\ln\ell = 0.1 : 0.1 : 3.2$				
Kernel	1day	2days	3days	4days	5days	1day	2days	3days	4days	5days
SE	0.127	0.143	0.157	0.168	0.179	0.180	0.186	0.193	0.201	0.208
RQ	0.125	0.142	0.157	0.168	0.178	0.152	0.159	0.167	0.174	0.182
PERD	0.262	0.279	0.325	0.348	0.347	0.193	0.203	0.222	0.234	0.239
NN	0.128	0.143	0.157	0.168	0.178	0.177	0.176	0.179	0.183	0.189
POLY	0.125	0.143	0.157	0.168	0.178	0.135	0.146	0.158	0.169	0.179
Gabor	0.126	0.144	0.158	0.169	0.180	0.166	0.158	0.170	0.180	0.185
MATRN1	0.129	0.142	0.156	0.167	0.177	0.129	0.142	0.156	0.167	0.177
MATRN2	**0.125**	**0.142**	**0.156**	**0.167**	**0.177**	**0.125**	**0.142**	**0.156**	**0.167**	**0.177**

the historical load corresponding to each time interval. The tables are organized so that two values of the initialization of $\ln\ell$ are provided for a range of the length of the training interval for the various kernels tested. The first row of these tables denotes two cases of initialization of $\ln\ell$. From column 2 to column 6 corresponds to the first case when $\ln\ell = \{0.1, 0.2, \cdots, 1\}$. From column 7 to column 11 corresponds to the second case when $\ln\ell = \{0.1, 0.2, \cdots, 3.2\}$. The second row of these tables denotes the length of the training data, which ranges from previous 1-day's data up to previous 5-days'. The first column of these tables is the name of the eight kernels. Inside the tables are the mean square errors (MSE) of the day-ahead load forecasting. Let $\hat{y}^k_{n,1}$ be load forecasting at the nth day at time point 1 with $\ln\ell = k$ for some year. Then, $\hat{\mathbf{y}}^k_n = (\hat{y}^k_{n,1}, \hat{y}^k_{n,2}, \cdots, \hat{y}^k_{n,96})^T$ is all the load forecasting at nth day in the same year. Let the $\mathbf{y}^k_n = (y^k_{n,1}, y^k_{n,2}, \cdots, y^k_{n,96})^T$ be the real load value at nth day corresponding to the $\hat{\mathbf{y}}^k_n$, then the MSE for this case at this year is computed as $\frac{1}{10K}\frac{1}{365-N}\sum_{k=0.1}^{K}\sum_{n=1}^{365-N}\frac{(\hat{\mathbf{y}}^k_n-\mathbf{y}^k_n)^T(\hat{\mathbf{y}}^k_n-\mathbf{y}^k_n)}{96}$, where $K = 1$ or $K = 3.2$ is the boundary of initialization of $\ln\ell$ and the $N \in \mathbb{N}, N \in [1, 5]$ is the length of the training day. For each

Table 3.3: forecasting MSE for kernels with different length of training data at 2014

	intialization $\ln\ell = 0.1 : 0.1 : 1$					intialization $\ln\ell = 0.1 : 0.1 : 3.2$				
Kernel	1day	2days	3days	4days	5days	1day	2days	3days	4days	5days
SE	4.233	5.451	6.164	6.664	6.951	4.252	5.467	6.184	6.682	6.965
RQ	4.225	5.418	6.122	6.613	6.899	4.229	5.418	6.121	6.612	6.898
PERD	6.959	7.861	8.729	9.003	9.443	5.092	6.200	6.960	7.387	7.721
NN	4.241	5.445	6.161	6.657	6.947	4.289	5.482	6.202	6.703	6.993
POLY	4.229	5.438	6.148	6.648	6.934	4.229	5.438	6.147	6.647	6.933
Gabor	4.238	5.486	6.207	6.710	6.999	4.455	5.646	6.345	6.845	7.120
MATRN1	**4.223**	**5.394**	**6.096**	**6.591**	**6.877**	**4.223**	**5.394**	**6.096**	**6.591**	**6.877**
MATRN2	4.223	5.408	6.110	6.605	6.892	4.223	5.408	6.110	6.605	6.892

Table 3.4: forecasting MSE for kernels with different length of training data at 2015

	intialization $\ln\ell = 0.1 : 0.1 : 1$					intialization $\ln\ell = 0.1 : 0.1 : 3.2$				
Kernel	1day	2days	3days	4days	5days	1day	2days	3days	4days	5days
SE	3.841	4.569	4.837	4.924	4.886	3.855	4.577	4.847	4.935	4.897
RQ	3.832	4.533	4.791	4.877	4.797	3.833	4.533	4.791	4.877	4.810
PERD	6.625	7.169	7.457	7.340	7.458	4.715	5.377	5.650	5.673	5.685
NN	3.847	4.560	4.835	4.925	4.888	3.898	4.603	4.886	4.984	4.955
POLY	3.836	4.556	4.820	4.906	4.871	3.836	4.555	4.820	4.906	4.871
Gabor	3.850	4.608	4.881	4.974	4.938	4.037	4.740	5.001	5.096	5.058
MATRN1	**3.830**	**4.514**	**4.773**	**4.860**	**4.824**	**3.830**	**4.514**	**4.773**	**4.860**	**4.824**
MATRN2	3.830	4.524	4.783	4.870	4.836	3.830	4.524	4.783	4.870	4.836

table and kernel, the MSE increases with the number of training days' data. With the previous one day's data as the training data, the kernels have the "best" forecasting accuracy. This means, in general, the feature of load data in the previous day has the strong similarity as the following day's load data. In addition, it may be noted that the most of kernels' forecasting accuracy decreases with the larger hyperparameter $\ln\ell$. With the boundary of the initialization of hyperparameter $\ln\ell$ increased from 1 to 3.2, the kernels have a larger MSE except for the kernels PERD, MATRN1 and MATRN2, of these, the PERD has the biggest the MSE. The reason why the forecasting accuracy is decreased for the most of the kernels here with the bigger range of hyperparameter $\ln\ell$ may due to the local maxima in their marginal likelihood function at large $\ln\ell$. The Matérn kernals yield the best forecasting accuracy which is also robust to the initialization ranges of the hyperparameter $\ln\ell$. Thus, the Product of Matérn kernels (PMK), which is expressed in (3.21) (MTN2*MTN2), is proposed for the residential area data. The Additive of Matérn kernels (AMK), which is expressed in (3.22), is employed to learn the business area data, because these composite Matérn kernels are

slightly better than the single Matérn kernels.

3.3.3 Training Data Selection

Since the temperatures and the load are varied daily, it may improve the forecasting accuracy if the relationship between the temperature and load demand can be used [121]. In this section, an algorithm is proposed to select the training data, with which the CMKs increase the forecasting accuracy to a great degree.

The general idea of the algorithm is to use the daily maximum and minimum temperatures as the indices to select the historical training data. A historical date whose daily maximum and minimum temperatures are close to the predicting day's maximum and minimum temperatures is found, then, the corresponding historical load data is taken as the training data for the kernels.

To increase the search record and load patterns, the previous year's daily maximum and minimum temperatures from [88] are added. The algorithm is summarized in the **Algorithm 1**.

Algorithm 1 training data selection

1: **Input**: $m \in \mathbb{N}$, the number of days data used for kernel training; $N \in \mathbb{N}$, the number of day in the historical data and $N > m$

2: **Output**: $\mathbf{h}$, the vector of load data used for the kernel training

3: Initialization: let (x_p, y_p) as the maximum and minimum temperature of prediction day; let $\mathbf{x}_h = (x_1, x_2, ..., x_N)^T$ and $\mathbf{y}_h = (y_1, y_2, ..., y_N)^T$ be the historical daily maximum and minimum temperature vector respectively. Let $j = k = 1$, $\mathbf{u}$ is empty.

4: Compute the temperature difference vector $\mathbf{d} = (d_1, d_2, ..., d_N)^T$, where $d_t = \sqrt{(x_t - x_p)^2 + (y_t - y_p)^2}, t \in [1, N]$. Let $d_{bd} = d_N$, set the temperature difference boundary as prediction day and one day before prediction day.

5: Sort $\mathbf{d}$ from small to big and let $\mathbf{s}$ equal the sorted $\mathbf{d}$, with $\mathbf{s} = (s_1, s_2, ..., s_N)^T, s_i < s_{i+1}$. Let $\mathbf{R} = (r_1, r_2, ..., r_N)$ be the day indexes of $\mathbf{s}$. ▷ e.g.: if $s_1 = d_3$, then $r_1 = 3$

Algorithm 1 Part 2

```
while j <= m do
    if s_k <= d_bd then
        Add r_k th day load data to u; j ← j + 1
    end if
    k ← k + 1
    if k > N then
        len ← the vector length of u
        break
    end if
end while
if len == 0 then
    h ← load data from yesterday
else
    h ← u
end if
```

To select the right number of day m in the Algorithm 1, the exhausted test has been done by ranging $m = [2, 3, ..., 20]$. The results are summarized as figure 3.5.

From figure 3.5 below, the upper plot displays the results of varies m for the residential area at 2012 and 2013, and the lower plot shows the results for the business area at 2014 and 2015. When $11 < m < 20$, the MSE is decreased to a stable level, which provides a range of candidate values for the m.

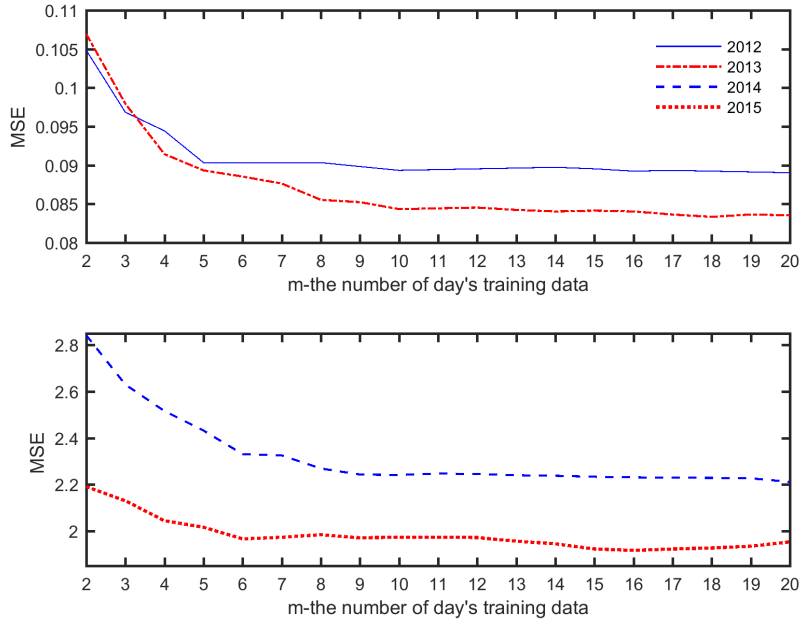

Figure 3.5: the training data length m with the MSE

3.3.4 Day-ahead load forecasting result

According to the analysis in section 3.3.2, the weekday and weekend have clearly different patterns of load demand in the business area which is not so in the residential area. To test if an improvement can be made in the load forecasting during the weekend, the historical weekends' data is separated from weekdays for these four years, which is used for the training. Then, the algorithm 1 and the same CMKs are utilized to generate the day-ahead load forecasting only for the weekends. The overall results are summarized in table 3.5.

Table 3.5: MSE comparison for data selection

Training data	MSE at 2012	MSE at 2013	MSE at 2014	MSE at 2015
Previous one day	0.128	0.125	4.223	3.830
Data selection without weekend separated	0.089	0.084	2.213	1.955
Data selection with weekend separated	**0.084**	**0.081**	**1.823**	**1.559**

From Table 3.5, with the training data selection, the CMKs make a big improvement in the day-ahead load forecasting. When the weekend training data is treated separately, the CMKs make

another improvement, especially in the business area, which also matches the previous analysis that there is clearly load demand pattern difference between the weekday and the weekend in the Texas area.

The CMKs of the GPR provides an acceptable accuracy for day-ahead time series forecasting. The Yearly Mean Absolute Percentage Error (YMAPE) is defined as:

$$\text{YMAPE} = \frac{\text{Mean Absolutely Forecasting Error (MAFE)}}{\text{Yearly Mean Load (YML)}} \tag{3.23}$$

The YMAPE for each year is presented in Table 3.6. In the Albuquerque area, the YMAPE for 2012 and 2013 are less than 0.07 and in the north central of Texas area at 2014 and 2015 are less or equal to 0.03. However, there are some cases in both areas that the CMKs do not fit well because the historical data is insufficient to make a reasonable inference. In addition, during the forecasting day, even though there are differences identified between the predicted load and real load curves, such as the cases in the figures 3.6 to 3.9 (Note: the residential area data is filtered by the Savitzky-Golay (SG) filter in the figure), the CMKs with the selected training data can't tune the curve and rectify the forecasting. Thus, a promising method to solve this shortcoming is to learn the forecasting error and then gradually adjust the predicted curve. Meanwhile, the advantage of the CMKs of GPR, in some sense, is kept. In other words, the method may not only need to have the good very short-term forecasting ability according to observed forecasting error, but also preserve the curve shape and short-term forecasting ability from the CMKs of GPR.

Table 3.6: Yearly mean average percent error for the CMKs of GPR

Index	2012 PMK	2013 PMK	2014 AMK	2015 AMK
MSE	0.084	0.081	1.823	1.559
MAFE	0.206	0.207	1.008	0.939
Yearly Mean Load (YML)	3.035 MW	3.006 MW	45.502 Kwh	45.703 Kwh
YMAPE	**6.79%**	**6.89%**	**2.22%**	**2.05%**

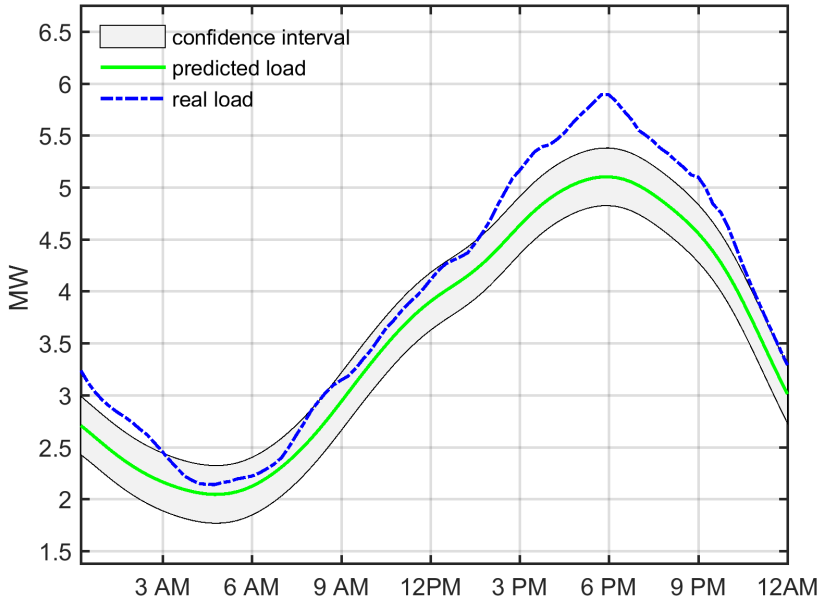

Figure 3.6: An anomalous case that the CMKs has a low performance in 2012 summer

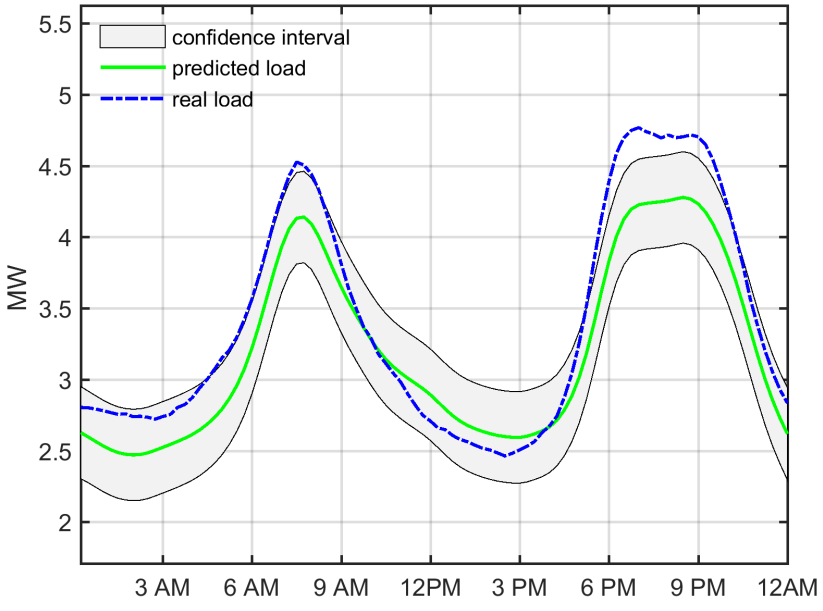

Figure 3.7: An anomalous case that the CMKs has a low performance in 2013 winter

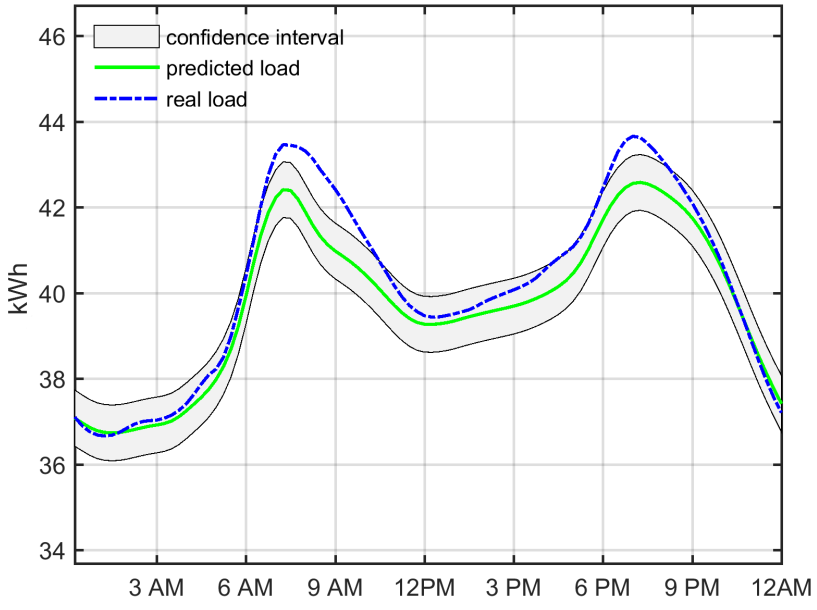

Figure 3.8: An anomalous case that CMKs has a low performance in 2014 winter

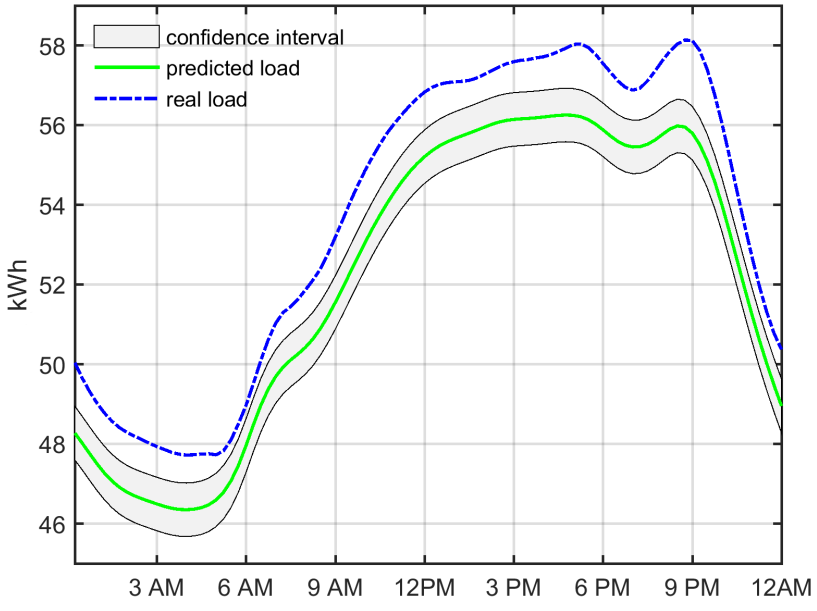

Figure 3.9: An anomalous case that CMKs has a low performance in 2015 summer.

3.4 The Curve Tuning Algorithm

3.4.1 Methodology

The main idea of proposed method is taking the new library to approximate the real load demand curve and forecasting gradually with time by using the K-SVD. This algorithm can be divided into three steps: first, when the error, generated by the forecasting, surpasses some boundary at certain time t, taking the K-SVD to "decompose" the outputs from the CMKs so to build a new library. Second, try to approximate known real load curve (real load demand curve from time 1 up to time t) by using the K-SVD based on the same time length of the generated dictionary. With coefficients, $\mathbf{k}$, generated from the K-SVD, generating the tuning curve with all day length of the same dictionary; Last, select the best tuned curve for certain T_0 in equation 3.15. The detailed explanation is introduced below.

Assume at tth time point of nth day for a certain year, where $t \in [5, 96]$, (Note: the simulation data sample is 15-min interval, so $15 \times 96 = 24$ hours and take first 4 time points to compute the accumulated forecasting error, $n \in [1, 365]$, both $t, n \in \mathbb{N}$, the CMKs forecasting, $\hat{\mathbf{y}}_n \in \mathbb{R}^{96\times 1}$, has the forecasting error that surpasses the boundary $errB$, then,

Step1: Design New Dictionary

In the dictionary learning algorithms (DLA), the choice of the dictionary plays an important role in the derivation of the sparse representation. With an appropriate overcomplete dictionary, the DLA find the sparse representation of the target signal efficiently. Usually, an overcomplete dictionary, $\mathbf{D}$, is designed by adapting the content to approximate a given set of signal or chosen as a specific set of functions [120]. The proposed method takes previous one year and current year historical load demand data until $(n-1)th$ day as the old dictionary $\mathbf{D}_{old} \in \mathbb{R}^{96\times(365+n-1)}$ to generate the new overcomplete dictionary $\mathbf{D}_{new}$. The K-SVD is applied to solve the following problem:

$$\min_{\mathbf{D}_{old},\mathbf{k}} \left\{ ||\hat{\mathbf{y}}_n - \mathbf{D}_{old}\mathbf{k}||_F^2 \right\}, \quad \text{subject to} \quad ||\mathbf{k}||_0 = T_0 \tag{3.24}$$

where T_0 is the nonzero element in $\mathbf{k}$. To provide the best estimation of the real load demand $\mathbf{y}_n$ in **step2** and **step3**, the T_0 is varied in a range, which need to be tested on the data. Here, $T_0 \in [3, 16]$ and $T_0 \leq t - 3$. For a certain T_0, the $\mathbf{D}_{new}$ is built by the **Algorithm 2**:

Algorithm 2 build new library

1: **Input**: T_0, $\mathbf{D}_{old}$, $\hat{\mathbf{y}}_n$
2: **Output**: $\mathbf{D}_{new}$
3: Initialization: $index \leftarrow 1$, $\mathbf{D}_{new}$ is empty.
4: **while** $index \leq 365 + n - T_0$ **do**
5: Let set $\{k_i\}_{i \in \mathbf{v}}$, $\mathbf{v} \subset [index, 365 + n - 1]$, $||\mathbf{v}||_0 = T_0$, be the solution of equation 3.24
6: $\mathbf{D}_{new} \leftarrow (\mathbf{D}_{new}, \mathbf{D}_{old}(:, \mathbf{v}))$
7: $index \leftarrow \min(\mathbf{v}) + index$
8: **end while**
9: $\mathbf{D}_{new} \leftarrow$ unique atoms$(\mathbf{D}_{new})$

The underlying idea to estimate the real load demand curve by designing a new dictionary is: assume that the $\hat{\mathbf{y}}_n$ is a good estimation of the real load demand curve $\mathbf{y}_n$, which is already shown in the previous section. If $\mathbf{D}_{new}$ can be used to generate a sparse representation of $\hat{\mathbf{y}}_n$, it would also do to $\mathbf{y}_n$. Another benefit is during the new dictionary designing, the good forecasting $\hat{\mathbf{y}}_n$ is "decomposed" as the atoms in the $\mathbf{D}_{new}$, which keeps the advantage of the CMKs outputs $\hat{\mathbf{y}}$. In addition, comparing using $\mathbf{D}_{old}$ in Step 2, the $\mathbf{D}_{new}$ provides less likely to generate local minimum atoms during the forecasting. To make the $\mathbf{D}_{new}$ is overcompleted, the *while* loop in the algorithm 2 is designed to exhausted search and add more atoms in the $\mathbf{D}_{new}$, which inevitably add some noisy atoms, especially after the best T_0 is found by K-SVD. It is intuitive to delete the noise atoms here to solve this issue. Rather than that, this work tries to delete them in the **Step2** in order to keep the overcompleteness of $\mathbf{D}_{new}$. Note, you may consider filtering the $\mathbf{D}_{old}$ before applying **Algorithm 2**, in case its original atoms are noise.

Step2: Generate tuned curves

The main task in this step is to fit the partially known real load demand curve with $\mathbf{D}_{new}$ and

take the coefficients from the fitting, together with the whole day length of atoms from $\mathbf{D}_{new}$, to forecast the rest of day for each T_0. Before fitting the known real load, the $\mathbf{y}_n$ may be filtered by SG filter in case the partial of target signals are noise. Assume $\mathbf{D}_{new} \in \mathbb{R}^{96\times\ell}, \ell \geq T_0, \ell \in \mathbb{N}$, Let $\mathbf{y}_n^t \in \mathbb{R}^{t\times 1}, \mathbf{y}_n^t = \mathbf{y}_n(1:t)$ be the partially known load demand at tth time point of nth day, $\mathbf{D}_{new}^t$ is partial dictionary of $\mathbf{D}_{new}, \mathbf{D}_{new}^t = \mathbf{D}_{new}(1:t,:)$. The fitting basically uses K-SVD to solve following problem:

$$\min_{\mathbf{D}_{new}^t,\mathbf{k}} \left\{ ||\mathbf{y}_n^t - \mathbf{D}_{new}^t\mathbf{k}||_F^2 \right\}, \quad \text{subject to} \quad ||\mathbf{k}||_0 = T_0 \tag{3.25}$$

Let set $\mathbf{k}_s = \{k_i\}_{i\in\mathbf{v}}$, $\mathbf{v} \subseteq [1,\ell], ||\mathbf{v}||_0 = T_0$, and $\mathbf{D}_{new}^s$ be the first solution. Then let $\mathbf{g} = (1,1,\cdots,1)^T, \mathbf{g} \in \mathbb{R}^{96}$, the first forecasting can be written as:

$$\begin{aligned} \mathbf{H} = \mathbf{D}_{new}^t(:,\mathbf{v}) \oslash \mathbf{D}_{new}^s(:,\mathbf{v}),\ \mathbf{r}_D = \bar{\mathbf{H}} \\ \hat{\mathbf{y}}_{n,t_1} = (\mathbf{D}_{new}(:,\mathbf{v}) \oslash (\mathbf{g}\mathbf{r}_D))\mathbf{k}_s \end{aligned} \tag{3.26}$$

The main idea for equations 3.25 and 3.26 is: at tth time point, if the forecasting is not good, it learns the current real load curve to tune the forecasting. The learning process is actually using the K-SVD to fit the $\mathbf{y}_n^t$, finding the coefficients $\mathbf{k}$ to combine the atoms from $\mathbf{D}_{new}^t$. The tuning action basically means using nonzero coefficients of $\mathbf{k}$, $\mathbf{k}_s$, and transformed $\mathbf{D}_{new}$ to predict the load. This is based on the assumption that if the $\mathbf{k}_s$ and $\mathbf{D}_{new}^s$ fit $\mathbf{y}_n^t$ well, the $\mathbf{k}_s$ and $\mathbf{D}_{new}(:,\mathbf{v})$ may also fit the $\mathbf{y}_n$ well. You will see in the results part that this assumption actually produces high accuracy at very short term forecasting. Note, the *Hadamard division*, $\oslash$, in (3.26) is necessary because the K-SVD changes $\mathbf{D}_{new}^t$ internally and normalizes it as well. Usually, the first forecasting $\hat{\mathbf{y}}_{n,1}$ is barely enough to generate the best whole day load forecasting. The $while$ loop, similar to it in the **Algorithm 2**, is utilized with $index \leq \ell - T_0 + 1$. Let $\hat{\mathbf{Y}}_{n,t} = (\hat{\mathbf{y}}_{n,t_1}, \hat{\mathbf{y}}_{n,t_2}, ..., \hat{\mathbf{y}}_{n,t_m})$ be the forecasting set of the $while$ loop, where $m \leq 30$ is set in both areas. To select the best prediction $\hat{\mathbf{y}}_{n,t}$ at tth time point of nth day from $\hat{\mathbf{Y}}_{n,t}$ for T_0, the Curve Selection Model (**Algorithm 4**) is designed, which is introduced in the section 3.4.2. This step is summarized in the **Algorithm 3**

Algorithm 3 generate tuned curves

Input: T_0, $\mathbf{D}_{new}$, t, $\mathbf{y}_n^t$, $\hat{\mathbf{Y}}_{n,t}^s$
Output: $\hat{\mathbf{y}}_{n,t}$
Initialization: $index \leftarrow 1, m \leftarrow 1$
while $index \leq \ell - T_0 + 1$ **do**
    Let $\mathbf{k}_s \leftarrow \{k_i\}_{i\in\mathbf{v}}$, $\mathbf{v} \subseteq [index, \ell], ||\mathbf{v}||_0 = T_0, \mathbf{D}_{new}^s$, be the solution of (3.25)
    $\mathbf{H} = \mathbf{D}_{new}^t(:, \mathbf{v}) \oslash \mathbf{D}_{new}^s(:, \mathbf{v}),\ \mathbf{r}_D = \overline{\mathbf{H}}$
    $\hat{\mathbf{y}}_{n,t_m} \leftarrow (\mathbf{D}_{new}(:, \mathbf{v}) \oslash (\mathbf{gr}_D))\mathbf{k}_s$
    $\hat{\mathbf{Y}}_{n,t}(:, m) \leftarrow \hat{\mathbf{y}}_{n,t_m}$ and $\mathbf{K}_{n,t}^s(:, m) \leftarrow \mathbf{k}_s$
    $m \leftarrow m + 1,\ \ index \leftarrow \min(\mathbf{v}) + index$
    if $m > 30$ **then**
        break
    end if
end while
$\hat{\mathbf{y}}_{n,t} \leftarrow$ **Algorithm 4**$\big(\hat{\mathbf{Y}}_{n,t}, \mathbf{K}_{n,t}^s, \hat{\mathbf{Y}}_{n,t}^s, \text{varargin}\big)$

Step3: Select tuned curve

From Step 1 to Step 2, a real load forecasting at the tth time point is generated, $\hat{\mathbf{y}}_{n,t}$, for a certain length of T_0. Since the best T_0 is varied with the length of t, instead of selecting one T_0 to do the load curve tuning and forecasting, a range, $T_0 \in [3, 16]$ and $T_0 \leq t - 3$, are introduced. Therefore, there will be a $T_0 - 2$ or 14 tuned curves. Let $\hat{\mathbf{Y}}_{n,t}^s = (\hat{\mathbf{y}}_{n,t}^1, \hat{\mathbf{y}}_{n,t}^2, ..., \hat{\mathbf{y}}_{n,t}^{T_0-2})$, then the main task in this step is to select a $\hat{\mathbf{y}}_{n,t}^i, i \in [1, T_0 - 2]$ that is considered to be the best.

Since during the Curve Selection Model in the Step 2, the predicted tuned curves are all taken as the first round selection, the $\hat{\mathbf{Y}}_{n,t}^s$ is considered to be a good candidate set here. The ideas to choose $\hat{\mathbf{y}}_{n,t}^i$ is: let $\mathbf{m} = \overline{\hat{\mathbf{Y}}_{n,t}^s}$, $\mathbf{m}_s = \mathbf{m} \setminus \{\max(\mathbf{m}), \min(\mathbf{m})\}$, and $m_{all} = \overline{\mathbf{m}}_s$. The $\hat{\mathbf{y}}_{n,t}^i$ is selected when its mean value $\overline{\hat{\mathbf{y}}_{n,t}^i} = m_t$, where $|m_t - m_{all}| = \min(|\mathbf{m}_s - m_{all}|)$. To make the prediction is improved overtime t, a similar way is applied to select the final tuned curve. Let $\mathbf{C}_p = \big(\hat{\mathbf{y}}_{n,t-4}^i, ..., \hat{\mathbf{y}}_{n,t-1}^i\big), \mathbf{C} = (\mathbf{C}_p, \hat{\mathbf{y}}_{n,t}^i)$, and the final predicted curve at t is $\hat{\mathbf{y}}_{n,t}^{final}$ that equals

the row mean of $\mathbf{C}_s$, where $\overline{\mathbf{C}_s} = \overline{\mathbf{C}} \setminus \{\max(\overline{\mathbf{C}}), \min(\overline{\mathbf{C}})\}$.

3.4.2 Curve Selection Model

The main task for this model is for each T_0, to select a $\hat{\mathbf{y}}_{n,t_i}, i \in [1, m]$, as $\hat{\mathbf{y}}_{n,t}$, from $\hat{\mathbf{Y}}_{n,t} = (\hat{\mathbf{y}}_{n,t_1}, \hat{\mathbf{y}}_{n,t_2}, ..., \hat{\mathbf{y}}_{n,t_m})$ in the Step 2. This model can be generally divided into two parts:

(a). Design error feedback linear regression models

Before introducing the regression models, two important parameters, the variance of $\mathbf{k}_s$ ($\mathrm{Var}(\mathbf{k}_s)$) in Step 2 and the mean $\hat{\mathbf{y}}_{n,t_i}$, are designed to further select the predicted curves/atoms. After solving (3.25), each $\hat{\mathbf{y}}_{n,t_i}$ has an unique $\mathbf{k}_s$, and each $\mathbf{k}_s$ takes the atoms from the $\mathbf{D}^s_{new}$ to fit the target signal $\mathbf{y}^t_n$. In the **Algorithm 3**, the ORMP algorithm takes the most similar atom to the target signal first, and atoms to their residual thereafter to the K-SVD. Then, the K-SVD continuously tunes the $\mathbf{k}_s$ and the corresponding atoms in the $\mathbf{D}_{new}$. Thus, when the atoms that are similar to the target signal are chosen, the K-SVD in the **Algorithm 3** tends to tune the left atoms and $\mathbf{k}_s$ heavily to fit the signal, which usually leads to deviated forecasting for the whole day. Therefore, the small $\mathrm{Var}(\mathbf{k}_s)$ is considered an important factor to show how K-SVD works during the fitting and how $\hat{\mathbf{y}}_{n,t_i}$ might be. However, there are some cases where the forecasting favors the K-SVD to tune the $\mathbf{k}_s$ and $\mathbf{D}^s_{new}$ heavily to some degree, especially when the daily load curve changes unexpectedly. Thus, the $\overline{\hat{\mathbf{y}}_{n,t_j}}$, as the second index, is designed. The forecasting is usually good when the predicted curve has the normal pattern and with the $\overline{\hat{\mathbf{y}}_{n,t_j}}$ close to the $\overline{\mathbf{y}_n}$. However, since the whole day $\mathbf{y}_n$ is not known yet (only known $\mathbf{y}^t_n$), it needs to forecast the $\overline{\mathbf{y}_n}$.

Two linear regression models (LRM) are designed with the known curve forecasting error as feedback to predict the $\overline{\mathbf{y}_n}$. The general idea is: assume $m \geq 8, t \geq 8, i \in [1, 11]$, compute the historical forecasting error, and the mean value of historical load forecasting with error feedback can be written as:

$$\mathbf{e}_{t-i} = \hat{\mathbf{y}}^i_{n,t-i}(t-i+1:t) - \mathbf{y}(t-i+1:t) \tag{3.27}$$

$$mY_{n,t-i} = \overline{\hat{\mathbf{y}}^i_{n,t-i}} - \overline{\mathbf{e}_{t-i}} \tag{3.28}$$

The linear regression models can be written as:

$$\mathbf{m}_{n,t} = (mY_{n,t-1}, mY_{n,t-2}, ..., mY_{n,t-11})^T$$
$$\hat{y}^{l1}_{n,t} = \mathbf{m}^T_{n,t}\boldsymbol{\beta}_1 + \epsilon_1; \hat{y}^{l2}_{n,t} = \mathbf{m}_{n,t}(1:k)^T\boldsymbol{\beta}_2 + \epsilon_2 \tag{3.29}$$

Where $\mathbf{m}_{n,t}$ is the predictors of the LRM. The $\hat{y}^{l1}_{n,t}$ and $\hat{y}^{l2}_{n,t}$, the forecasting of the $\overline{\mathbf{y}_n}$, are the responses of LRM, one with 11 historical time points (HTP) and the other with $k, k < 11$, HTP. The $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2$ are the coefficients and ϵ_1, ϵ_2 are the fitting residuals of these LRMs. k may need to be tested. Here, $k = 4$ for both areas.

The underlying idea is the model (3.29) focus on the daily mean load forecasting. It designs a interval, range$(\hat{y}^{l1}_{n,t}, \hat{y}^{l2}_{n,t})$, that the good $\overline{\hat{\mathbf{y}}_{n,t_j}}$ would fall in, otherwise choose the $\hat{\mathbf{y}}_{n,t_j}$ with the minimum $\mathrm{Var}(\mathbf{k}^j_s)$. Note, the error feedback in equations (3.27,3.28) forces the $\overline{\hat{\mathbf{y}}_{n,t_i}}$ to converge to the $\overline{\mathbf{y}_n}$ fast.

(b). Filter the noisy predicted curve/atom

To filter the noise atoms, historical records are used. Let $\mathbf{ld}_p, \mathbf{ld}_s \in \mathbb{R}^{(365+n-1)\times 1}$ be the daily peak load and daily minimum load records from previous $365 + n - 1$ days. The daily maximum peak load difference is defined: $maxLd_p = \max(\mathbf{dLd}^p_d)$, $\mathbf{dLd}^p_d = \mathrm{abs}(\mathbf{ld}_p(1:\mathrm{end}-1) - \mathbf{ld}_p(2:\mathrm{end}))$. The daily maximum minimum load difference $maxLd_s$ is defined in a similar way. To check the strange daily load pattern of $\hat{\mathbf{y}}_{n,t_j}$, six hours' mean load boundary set is defined: $\{mBd_i = \overline{\mathbf{m}_{\hat{\mathbf{Y}}^{s,T_0-3}_{n,t}(1+24(i-1):24i,:)}}\}, i \in [1,4]$, where $\mathbf{m}_{\hat{\mathbf{Y}}^{s,T_0-3}_{n,t}}$ is the mean value of previous predicted curve set for library: 3 to $T_0 - 1$. The $\{maxLd_p, maxLd_s\}$ defines the boundary of load curves changing from yesterday, and the $\{mBd_i\}$ defines the load curves change among different length of libraries.

This model is summarized in the **Algorithm 4**. In the **Algorithm 4**, t is separated by 8 because of building the LRM. The m is separated by 8 because of the tests. When $m \leq 8$, there are few atoms in the $\mathbf{D}_{new}$ to fit the $\mathbf{y}^t_n$. Due to the ORMP property, the atoms that are most similar to the $\mathbf{y}^t_n$ would be chosen first. Thus, $\hat{\mathbf{y}}_{n,t} = \hat{\mathbf{Y}}^f_{n,t}(:,1)$. During the tests, the $mDbd$ is recommended to set smaller in the morning hours than in the afternoon hours, to avoid irregular forecasting due to the CTA does not have enough information. For business area, $mDbd \in \{0.8, 2\}$, and $mDbd \in \{0.09, 0.5\}$ for the residential area.

Algorithm 4 curve Selection Model

Input: $n, t, T_0, \hat{\mathbf{Y}}_{n,t}, \mathbf{ld}_p, \mathbf{ld}_s, \mathbf{C}_{all}, \mathbf{K}^s_{n,t}, \hat{\mathbf{Y}}^s_{n,t}$
Output: $\hat{\mathbf{y}}_{n,t}$
$m \leftarrow$ column length $\hat{\mathbf{Y}}_{n,t}$
if $t \geq 8$ **then**
    if $m > 8$ **then**
        Compute $\hat{y}^{l1}_{n,t}, \hat{y}^{l2}_{n,t}$ using (3.27∼3.29)
        $len = \mathrm{round}(\frac{m}{2})$
        $\hat{\mathbf{Y}}_{n,t} = \hat{\mathbf{Y}}_{n,t}(:, 1:len)$
        $\mathbf{K}^s_{n,t} = \mathbf{K}^s_{n,t}(:, 1:len)$
        $\mathbf{S}_y \leftarrow \mathrm{sort}((\overline{\hat{\mathbf{Y}}_{n,t}}, \mathrm{Var}(\mathbf{K}_{n,t})), 2)$ ▷ sort $\overline{\hat{\mathbf{Y}}_{n,t}}$ by $\mathrm{Var}(\mathbf{K}_{n,t})$
        $listLT \leftarrow$ find $\mathbf{S}_y(:,1)$ in the $\mathrm{range}(\hat{y}^{l1}_{n,t}, \hat{y}^{l2}_{n,t})$
        if $\mathrm{notEmpty}(listLT)$ **then**
            $\hat{\mathbf{Y}}^c_{n,t} \leftarrow \hat{\mathbf{Y}}_{n,t}(:, listLT)$
            $indexMinS \leftarrow \min(\mathrm{abs}(\overline{\hat{\mathbf{Y}}^c_{n,t}} - \mathrm{mean}(\hat{y}^{l1}_{n,t}, \hat{y}^{l2}_{n,t})))$
            $\hat{\mathbf{y}}^s_{n,t} \leftarrow \hat{\mathbf{Y}}^c_{n,t}(:, indexMinS)$
        else
            $indexMinV \leftarrow \min(\mathrm{abs}(\mathbf{S}_y(:,1) - \mathrm{mean}(\hat{y}^{l1}_{n,t}, \hat{y}^{l2}_{n,t})))$
            $\hat{\mathbf{y}}^s_{n,t} \leftarrow \hat{\mathbf{Y}}_{n,t}(:, indexMinV)$
        end if
    else
        $\hat{\mathbf{y}}^s_{n,t} \leftarrow \hat{\mathbf{Y}}_{n,t}(:,1)$
    end if
else
    if $m \geq 8$ **then**
        $indexMinV \leftarrow \min(\mathrm{Var}(\mathbf{K}^s_{n,t}(:, \mathrm{round}(\frac{m}{2})))),\ \hat{\mathbf{y}}^s_{n,t} \leftarrow \hat{\mathbf{Y}}_{n,t}(:, indexMinV)$

Algorithm 4 Part 2

else
  $\hat{\mathbf{y}}^s_{n,t} \leftarrow \hat{\mathbf{Y}}_{n,t}(:, 1)$
end if
end if
Compute $maxLd_p, maxLd_s, mBd_i, \forall i \in [1, 4]$ as in section 3.4.2 (**b**)
if $\text{abs}(\max(\hat{\mathbf{y}}^s_{n,t}) - \mathbf{ld}_p(\text{end})) < maxLd_p$ and $\text{abs}(\min(\hat{\mathbf{y}}^s_{n,t}) - \mathbf{ld}_s(\text{end}) < maxLd_s$ **then**
  if $T_0 > 4$ **then**
    $crvMcur_i = \overline{\hat{\mathbf{y}}^s_{n,t}(1 + 24(i-1) : 24i)}, i \in [1, 4]$
    if $\text{abs}(crvMcur_i - mBd_i) < mDbd, \forall i \in [1, 4]$ **then**
      $\hat{\mathbf{y}}_{n,t} \leftarrow \hat{\mathbf{y}}^s_{n,t}$
    else
      $\hat{\mathbf{y}}_{n,t} \leftarrow \text{empty}()$
    end if
  else
    $\hat{\mathbf{y}}_{n,t} \leftarrow \hat{\mathbf{y}}^s_{n,t}$
  end if
else
  $\hat{\mathbf{y}}_{n,t} \leftarrow \text{empty}()$
end if

3.4.3 Curve Tuning Algorithm

The main framework of the CTA is designed in the **Algorithm 5**. Inside of the algorithm, the $errB$ is the MSE error boundary to start the CTA, which measures the fitting effects between the predicted curve and the real daily load curve. The $tTime$ is the time point to start the CTA. $tTime \geq 10$ is recommended so the algorithm has more real load curve data to "learn". $\mathbf{y}_{tuned}$ is the output of the final tuned curve and the $\mathbf{tm}_{tuned}$ is to record the number of times the CTA is launched. Both $\mathbf{y}_{tuned}$ and $\mathbf{tm}_{tuned}$ are used to measure the results in section 3.5. When the

tendency may not be seized correctly by the CTA, the tuned curve may need more freedom to change. Thus, in step 39, the tendency step is overwritten by the output from the best dictionary.

Algorithm 5 curve Tuning Algorithm

1: **Input**: $n, t, errB, tTime$, $\mathbf{D}_{old}, \hat{\mathbf{y}}_n, \mathbf{y}_n^t(\text{filtered}), \mathbf{C}_{all}$
2: **Output**: $\hat{\mathbf{y}}_{n,t}^{final}$, $\mathbf{C}_{all}$, For simulation test: $\mathbf{y}_{tuned}, \mathbf{tm}_{tuned}$
3: Initialization: $condTune = 0, \mathbf{tm}_{tuned} \leftarrow \text{zeros}(96, 1)$,$tuned = 0$
4: **if** not($tuned$) **then**
5: $\quad \mathbf{err}_{fit} \leftarrow \hat{\mathbf{y}}_n(1:t) - \mathbf{y}_n^t$
6: $\quad errFitNow \leftarrow \frac{\mathbf{err}_{fit}^T \mathbf{err}_{fit}}{t}$ ▷ Get MSE of fitting until current t
7: $\quad$ **if** $t \geq tTime$ **or** $t \geq 10$ **then**
8: $\quad\quad errFit10 \leftarrow \frac{\mathbf{err}_{fit}(t-9:t)^T \mathbf{err}_{fit}(t-9:t)}{t}$
9: $\quad\quad condTune \leftarrow (errFitNow > errB \,||\, errFit10 > errB)$
10: $\quad$ **end if**
11: $\quad$ **if** $t \geq tTime$ **then** ▷ When start to tune the curve, here $tTime = 10$
12: $\quad\quad \mathbf{tCurve}(1:t, 1) \leftarrow \hat{\mathbf{y}}_n(1:t)$
13: $\quad$ **end if**
14: $\quad \hat{\mathbf{y}}_{n,t}^{final} \leftarrow \hat{\mathbf{y}}_n$
15: $\quad \mathbf{C}_{all}(:, t) \leftarrow \hat{\mathbf{y}}_{n,t}^{final}$
16: **end if**
17: **if** $condTune$ **then**
18: $\quad tuned \leftarrow 1$
19: $\quad dicNum \leftarrow (16(t > 18) + (t - 2)(t \leq 18))$
20: $\quad$ **for** $T_0 \leftarrow 3, dicNum$ **do**
21: $\quad\quad \mathbf{D}_{new} \leftarrow$ **Algorithm 2**$(T_0, \mathbf{D}_{old}, \hat{\mathbf{y}}_n)$
22: $\quad\quad \hat{\mathbf{y}}_{n,t} \leftarrow$ **Algorithm 3**$(T_0, \mathbf{D}_{new}, t, \mathbf{y}_n^t, \hat{\mathbf{Y}}_{n,t}^s)$
23: $\quad\quad \hat{\mathbf{Y}}_{n,t}^s(:, T_0 - 2) = \hat{\mathbf{y}}_{n,t}$
24: $\quad$ **end for**
25: $\quad$ **if** exist('$\hat{\mathbf{Y}}_{n,t}^s$') **then**

Algorithm 5 Part 2

$\mathbf{m} \leftarrow \overline{\hat{\mathbf{Y}}^s_{n,t}(:, \text{find}(\hat{\mathbf{Y}}^s_{n,t}(1,:) > 0))}$ ▷ get the mean of nonzero vector of $\hat{\mathbf{Y}}^s_{n,t}$

$\mathbf{m}_s = \mathbf{m} \setminus \{\max(\mathbf{m}), \min(\mathbf{m})\}$

$m_{all} \leftarrow \overline{\mathbf{m}_s}$

$indexMin \leftarrow \min(|\mathbf{m}_s - m_{all}|)$

$\hat{\mathbf{y}}^i_{n,t} \leftarrow \hat{\mathbf{Y}}^s_{n,t}(:, \text{find}(\overline{\hat{\mathbf{Y}}^s_{n,t}} == \mathbf{m}_s(indexMin)))$

else

$\hat{\mathbf{y}}^i_{n,t} = \hat{\mathbf{y}}_n$

end if

$\mathbf{C}_p = \left(\hat{\mathbf{y}}^i_{n,t-4}, ..., \hat{\mathbf{y}}^i_{n,t-1}\right)$

$\mathbf{C} = (\mathbf{C}_p, \hat{\mathbf{y}}^i_{n,t})$

$\hat{\mathbf{y}}^{final}_{n,t} \leftarrow$ the row mean of $\mathbf{C}_s$, where $\overline{\mathbf{C}_s} = \overline{\mathbf{C}} \setminus \{\max(\overline{\mathbf{C}}), \min(\overline{\mathbf{C}})\}$

$\mathbf{C}_{all}(:, t) \leftarrow \hat{\mathbf{y}}^i_{n,t}$

if $\mathbf{tm}_{tuned}(t-1, 1) == 1$ **then**

$\hat{\mathbf{y}}^{final}_{n,t} \leftarrow \hat{\mathbf{y}}^i_{n,t}$ ▷ this works when tendency isn't correctly assumed

end if

$\mathbf{tCurve}(t, 1) \leftarrow \hat{\mathbf{y}}^{final}_{n,t}(t)$

$\mathbf{tm}_{tuned}(t, 1) \leftarrow 1$

$\mathbf{y}_{tuned} \leftarrow \mathbf{tCurve}$

end if

3.5 Results

In this section, three aspects' results are summarized. They are the very short term forecasting results (from 15-min to 4 hours), gradually tuning results, and the whole day tuning results.

3.5.1 Very Short Term Forecasting

In figures 3.10 and 3.13, daily very short term load forecasting is summarized according to different starting time points for each year. To show the results clearly, five starting time points, 2:00AM in dash black line, 7:30AM in red solid line, 12:30PM in cyan dash line, 3:00PM in blue solid line, and 5:30PM in green solid line, are plotted. Since the historical load data sample interval is 15-min for each point, so in these figures, the x axis starts at 15-min and up to 4 hours (240-min) with 15-min intervals.

The daily average MSE for each year (DAMSEY) is computed as by: assume the forecasting starts at time point $t_s, t_s \in \{8, 30, 50, 60, 70\}$, i.e.: (2:00AM, 7:30AM, 12:30PM, 3:00PM, and 5:30PM) of nth day, $n \in [1, 365]$, of a certain year. Let the predicted curve at this time point is $\hat{\mathbf{y}}_{n,t_s}^{final}$, the forecasting step size is $ss, ss \in [1, 16]$, i.e.: $ss \in [15min, 240min]$, then,

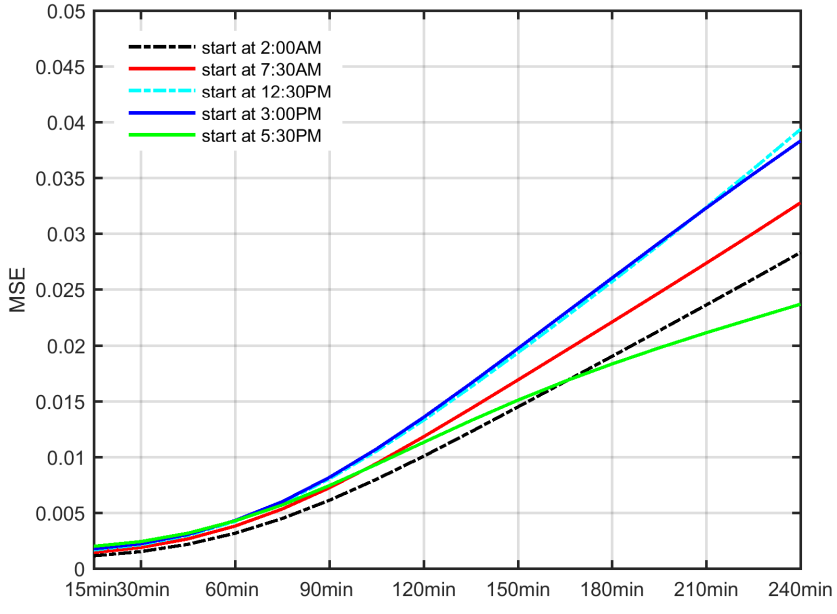

Figure 3.10: The daily average MSE for the Albuquerque area in 2012

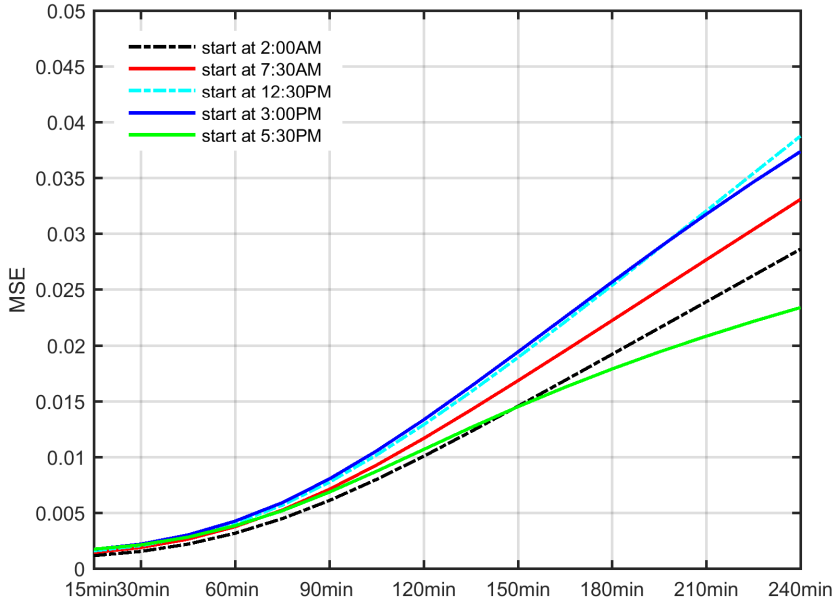

Figure 3.11: The daily average MSE for the Albuquerque area in 2013

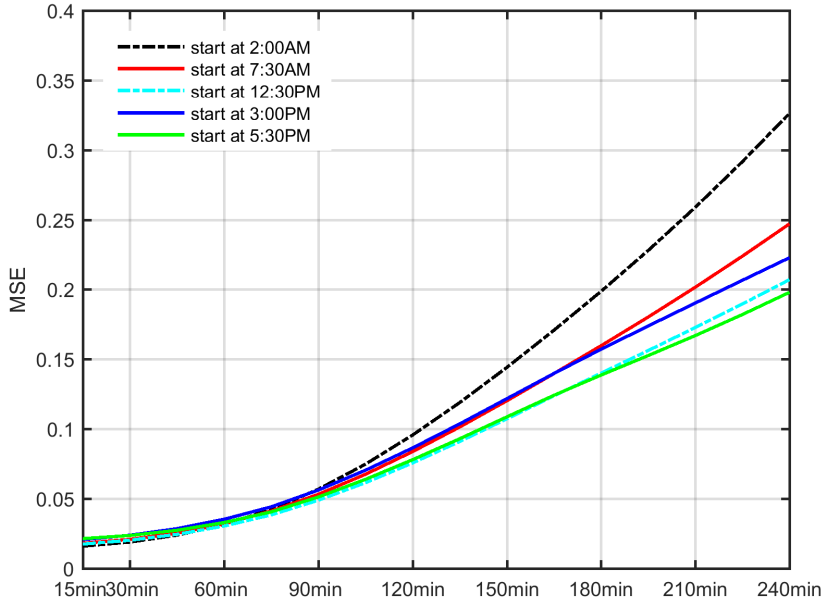

Figure 3.12: The daily average MSE for the north central Texas area in 2014

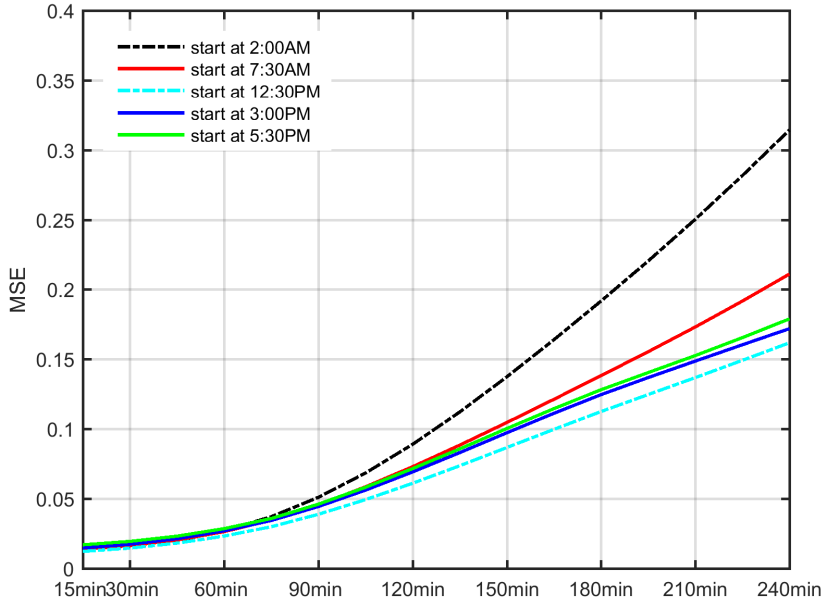

Figure 3.13: The daily average MSE for the north central Texas area in 2015

$$\mathbf{e}_{n,t_s} = \hat{\mathbf{y}}^{final}_{n,t_s} - \mathbf{y}_n \tag{3.30}$$

$$mse^{ss}_{n,t_s} = \frac{\mathbf{e}_{n,t_s}(t_s+1:t_s+ss)^T \mathbf{e}_{n,t_s}(t_s+1:t_s+ss)}{ss} \tag{3.31}$$

$$mmse^{ss}_{n,t_s} = \frac{1}{365}\frac{1}{96-ss-t_s+1}\sum_{n=1}^{365}\sum_{t=t_s}^{96-ss} mse^{ss}_{n,t} \tag{3.32}$$

Where $\mathbf{e}_{n,t_s}$ is the fitting error at t_s and the mse^{ss}_{n,t_s} is the MSE for the forecasting step size ss at time point t_s, and the $mmse^{ss}_{n,t_s}$ is the daily average MSE with the step size ss at the starting time point t_s in a year, which corresponds to one point of the line, which starts at time t_s in these figures.

It is clear to see in these figures, the proposed method has very high accuracy at very short term forecasting for both areas. With the Albuquerque data, no matter when it starts during the day, the DAMSEY is less than 0.005 (i.e.: YMAPE$<2\%$) for forecasting within one hour. For forecasting within 4 hours, the DAMSEY is less than 0.04. With the north central Texas data,

the DAMSEY is less than 0.04 (i.e.: YMAPE$< 0.3\%$) for forecasting within one hour and the DAMSEY is less than 0.34 for forecasting within 4 hours no matter when it starts. From the figures 3.10 to 3.13, the DAMSEY with the Texas area data is gradually decreasing in the morning, and it converges in the afternoon, while with the Albuquerque area data, the DAMSEY starts dropping quickly in the afternoon. Consequentially, with more daily load curve information available, the proposed method learns the known load curve and gradually improves the forecasting. Its property of gradually tuning is further investigated in the following section 3.5.2.

3.5.2 Gradually Tuning Property

The gradually tuning property of the proposed method is analyzed similarly to Section 3.5.1 but with the x axis as the time points for the forecasting rather than forecasting step size. In figures 3.14 and 3.17, daily gradually tuning MSE in a year (DGTMSE) is plotted for both areas. The blue dash lines are for the forecasting MSE of proposed method, the red line is for the GPR forecasting, and the green solid line is the result for partially combing the forecasting from proposed method with GPR's forecasting. To look at the gradually tuning property of proposed method, the MSE at the rest of day from the starting point (t_s) is computed. In other words, the forecasting step size is $ss = 96 - t_s$ for every test time point. The MSE of the GPR forecasting and the partially combined forecasting are measured in the same way.

The DGTMSE for these figures are computed as following: assume the forecasting starts at time point $t_s, t_s \in \{8, 12, 16, ..., 92\}$, i.e.: (2AM, 3AM, 4AM, ...,11PM) of nth day, $n \in [1, 365]$, of a certain year. Let the predicted curve for proposed method at this time point is $\hat{\mathbf{y}}_{n,t_s}^{final}$, the GPR forecasting is $\hat{\mathbf{y}}_{n,t_s}^{gpr}$, and the forecasting step size is $ss = 96 - t_s$, then the fitting error for the

proposed method is the same as equation 3.30,

$$\mathbf{e}_{n,t_s}^{gpr} = \hat{\mathbf{y}}_{n,t_s}^{gpr} - \mathbf{y}_n, \;\; \mathbf{e}_{n,t_s}^{comb} = \mathbf{e}_{n,t_s}(t_s+1:t_s+16) + \mathbf{e}_{n,t_s}^{gpr}(t_s+17:96) \tag{3.33}$$

$$mse_{n,t_s} = \frac{\mathbf{e}_{n,t_s}(t_s+1:96)^T \mathbf{e}_{n,t_s}(t_s+1:96)}{96-t_s} \tag{3.34}$$

$$gmse_{n,t_s} = \frac{1}{365}\sum_{n=1}^{365} mse_{n,t_s} \tag{3.35}$$

where $\mathbf{e}_{n,t_s}^{gpr}$ is the fitting error of the GPR forecasting, the $\mathbf{e}_{n,t_s}^{comb}$ is the fitting error of the partial combined forecasting. Note, in the DGTMSE computation for the partially combined case, the combined error is calculated only when $t_s \in \{8, 12, ..., 56\}$ for the Albuquerque data and $t_s \in \{8, 12, ..., 40\}$ for the Texas data and for the left time points, the error is generated by the proposed method. The mse_{n,t_s} is the MSE of forecasting generated by proposed method at t_s time point. The MSE of GPR and the partially combined case can be computed in the same way. The $gmse_{n,t_s}$ is the one point of the blue dash line for the proposed method, which has the DGTMSE at t_s time point for a whole year. The same way can be used to generate the DGTMSE for the GPR forecasting and the partially combined case.

It is not hard to see from the figures 3.14 to 3.17, the proposed method, as the dash blue line displayed, has the tendency to gradually reduce the DGTMSE during the day time. It crosses below the GPR (the red dash line) at 11 AM for the Albuquerque data and about 8 AM for the Texas data and its DGTMSE, thereafter, is rapidly decreased to the number that is close to the zero with the time going. However, its forecasting is not that good at the beginning of the day. This is mainly because the proposed method needs more data to learn at the beginning of the day. In other words, the proposed method needs more known load data during the day to find the $\overline{\mathbf{y}_n}$ to choose correct curve from the prediction curve set $\hat{\mathbf{Y}}_{n,t}$.

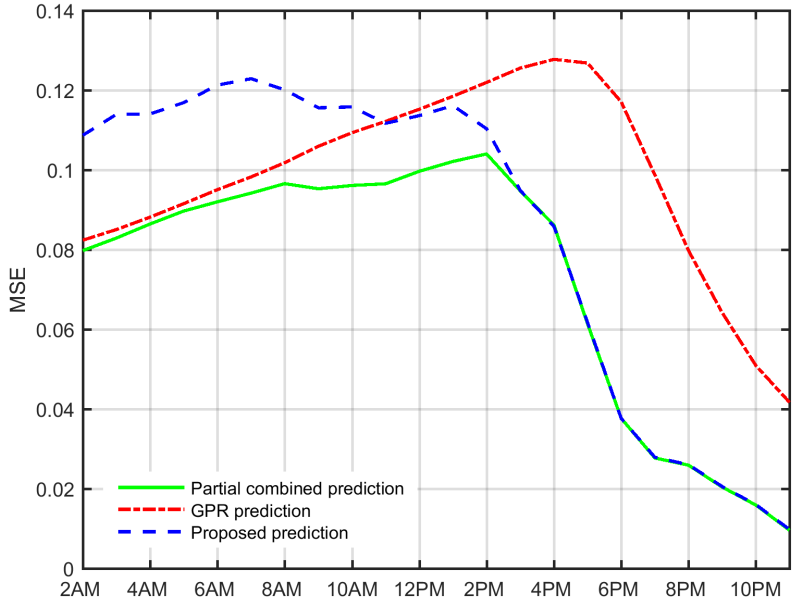

Figure 3.14: The daily gradually tuning MSE for the Albuquerque area in 2012

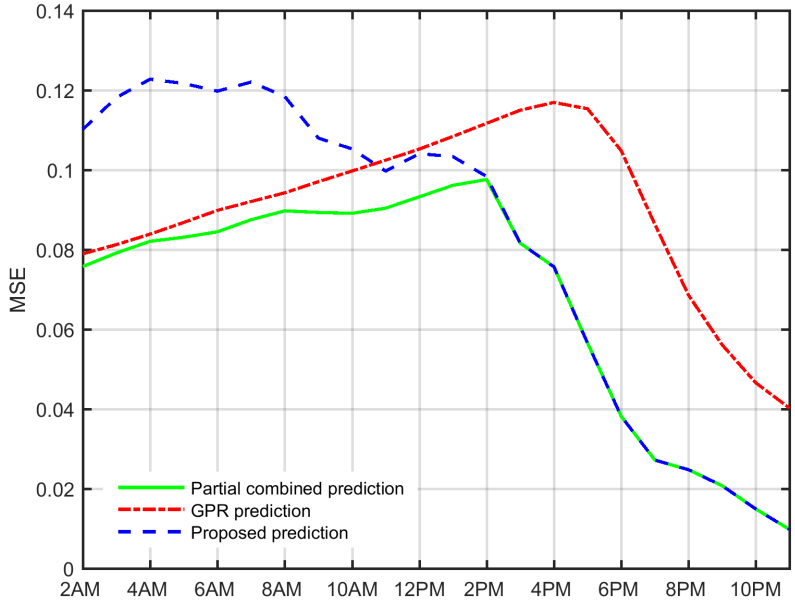

Figure 3.15: The daily gradually tuning MSE for the Albuquerque area in 2013

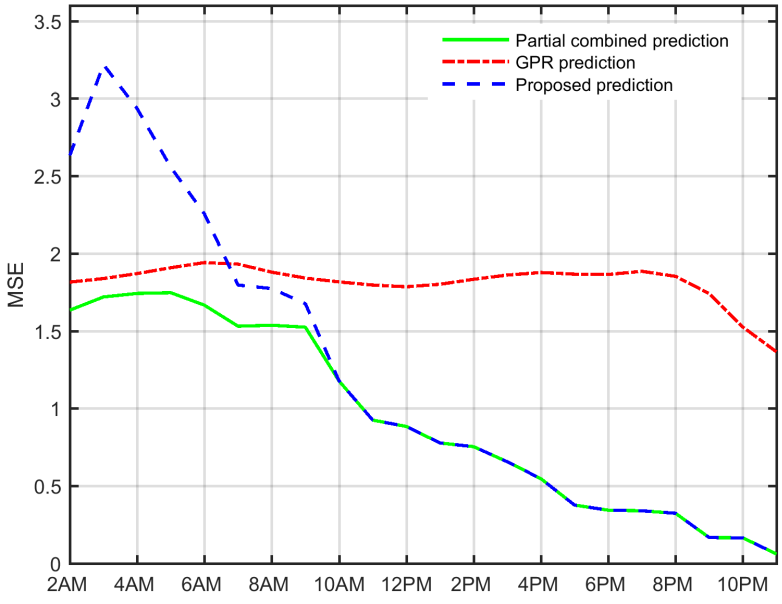

Figure 3.16: The daily gradually tuning MSE for the Texas area in 2014

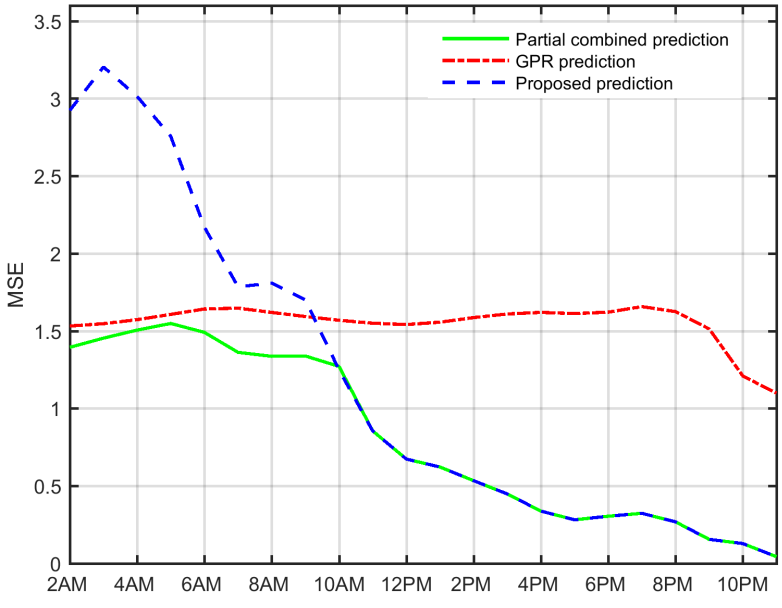

Figure 3.17: The daily gradually tuning MSE for the Texas area in 2015

Nevertheless, since the proposed method has a very high accuracy in the very short term fore-

casting (from 15-min to 4 hours), it is intuitive to partially combine the proposed method's forecasting with that from the GPR during the early time of the day. The combination is simple as equation (3.33) shows. It combines the most recent 4 hours' forecasting from the proposed method and rest of day's forecasting from the GPR. As the green lines show in figures 3.14 to 3.17, the partially combined way actually solves the issue of the proposed method and provides a decent forecasting for a whole day at the beginning of the day. Note, the time to end of this combination should be tested based on the data.

3.5.3 Whole Day Tuning result

In this part, the results of forecasting generated by the tuning algorithm are summarized. The generated curves are produced by the **Algorithm 5** as the vector $\mathbf{y}_{tuned}$ and the average tune time is 4 ($\mathbf{tm}_{tuned} = 4$). Corresponding to the figures 3.6 to 3.9, the real load curves, which the GPR can't fit well, are predicted again by the proposed method. The curves, generated by the proposed method plotted in red lines on figures 3.18 to 3.21, clearly forecasts better than the CMKs forecasting, with only a few times of tuning. The yearly GPR CMKs' forecasting and the tuning algorithm forecasting with SG filtered are then compared with MSE, RMSE, MAFE (Mean Absolutely Forecasting Error), and R^2, which are summarized in table 3.7. From all the year and all the indexes, the curve tuning algorithm surpasses the GPR in a great degree.

Table 3.7: Yearly Forecasting Accuracy Comparison

	CMKs				Proposed tuning Method			
Index	2012	2013	2014	2015	2012	2013	2014	2015
MSE	0.084	0.081	1.823	1.559	**0.016**	**0.017**	**0.296**	**0.276**
RMSE	0.290	0.285	1.350	1.249	**0.127**	**0.130**	**0.544**	**0.525**
MAFE	0.206	0.207	1.008	0.939	**0.105**	**0.105**	**0.395**	**0.379**
R^2	0.912	0.910	0.957	0.965	**0.983**	**0.981**	**0.993**	**0.994**
YMAPE	6.79%	6.89%	2.22%	2.05%	**3.46%**	**3.49%**	**0.87%**	**0.83%**

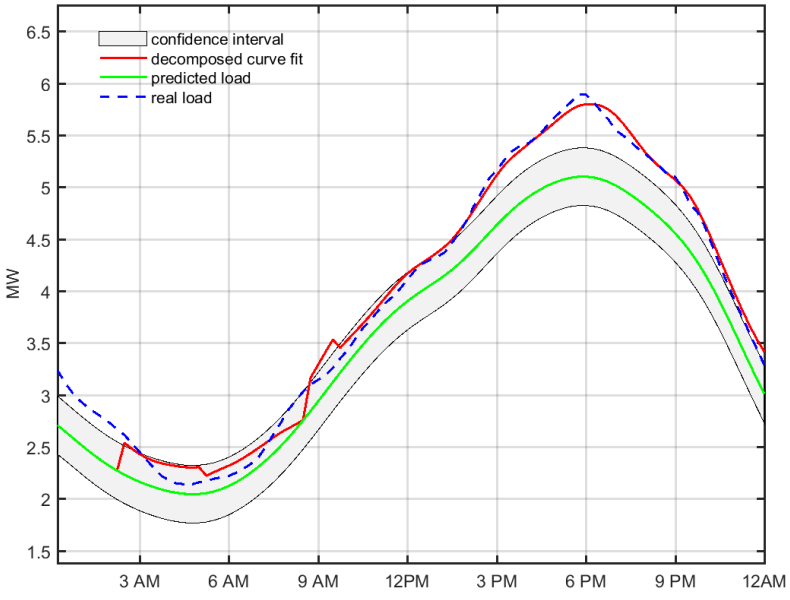

Figure 3.18: An example to show the performance of tuning algorithm in the summer 2012

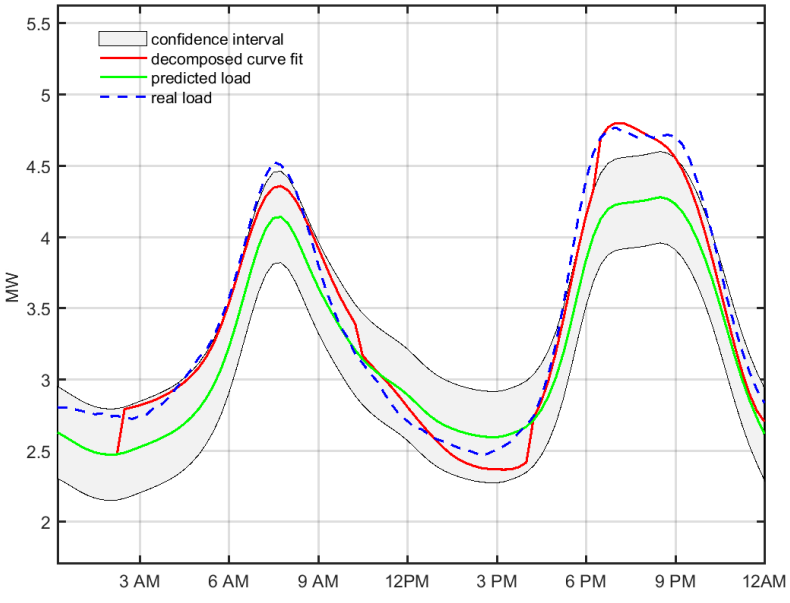

Figure 3.19: An example to show the performance of tuning algorithm in the winter 2013

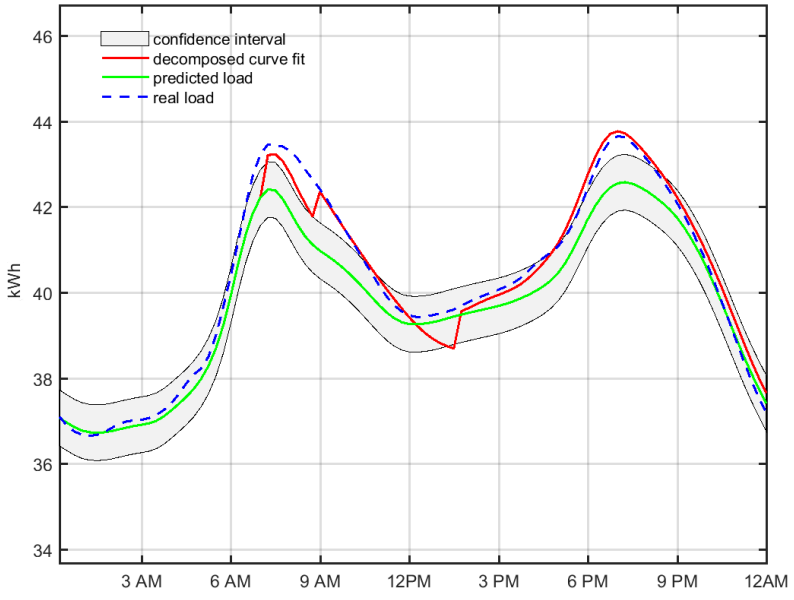

Figure 3.20: An example to show the performance of tuning algorithm in the winter 2014

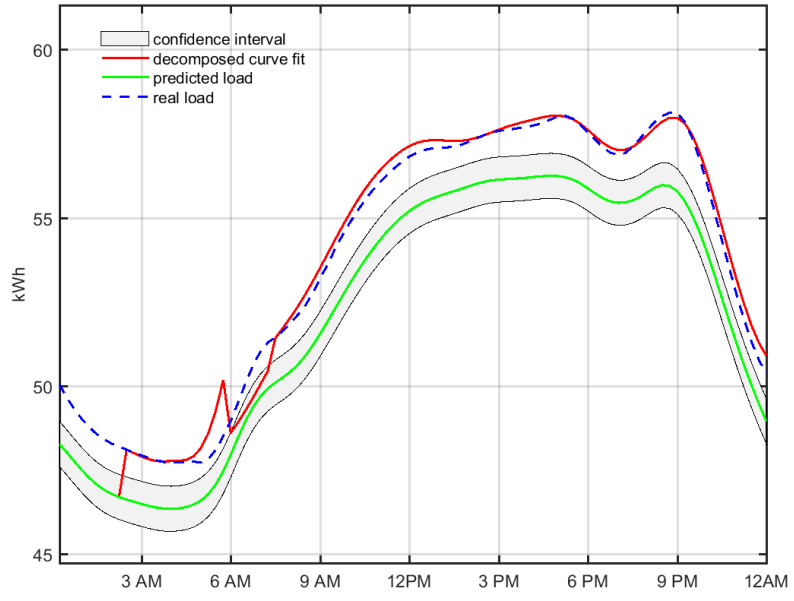

Figure 3.21: An example to show the performance of tuning algorithm in the summer 2015

Noticing the **Algorithm 5** is driven by the forecasting error, it can also be easily changed to a

user defined time driven, in which the tuning algorithm can be employed to improve the forecasting accuracy with the time.

The prediction of the mean daily load, i.e.: infer the $\overline{\mathbf{y}_n}$, may introduce the disturbance to the daily load forecasting in the early time, such as the blue dash lines in the figures 3.14 to 3.17 in the morning time, it also provides a parameter to accurately select the forecasting curve for the proposed method. If prior knowledge of the changing of $\overline{\mathbf{y}_n}$ could be seized in advance due to its correlated factors, such as temperature, human activities, and others, it may contribute to the proposed method to quickly converge the forecasting.

3.6 Conclusion

In this chapter, a novel algorithm regarding the short term and very short term distribution level load forecasting is proposed. It combines the composite kernel of GPR and the dictionary learning algorithm to do day-ahead load casting and daily load curve tuning.

To do the day-ahead load forecasting, two years of private load data from a the residential area and two years of public data from a business area are thoroughly analyzed. To select the best kernel in the framework of GPR, the commonly used the kernels are compared in detail. The Matérn kernels display the best forecasting ability and its forecasting accuracy is not sensitive to the initialization of its hyperparameter ℓ. Thus, it is chosen to build the composite kernel. To further improved the day-ahead load forecasting, an algorithm based on the maximum and minimum temperature is designed to select the training data for the composite kernels, which improves the composite kernels' forecasting accuracy.

Since in some cases, the composite kernel in the GPR framework is insufficient to deal with the very short-term forecasting with selected training data even there exists forecasting error, the curve tuning algorithm based on the K-SVD is designed. This algorithm, first, takes the K-SVD to decompose the forecasting from the composite kernels into a new dictionary, which inherits the advantage from the composite kernels in some sense. Then, it learns the known real load curve

by fitting it with the new dictionary using the K-SVD. With the result from the K-SVD, the whole day length of atoms are employed to forecast the daily real load curve. To select a good forecasting from the solution set, two indexes, the variance of the coefficients and the mean daily load forecasting, are employed. To forecast the mean daily load, an error feedback regression models are designed to stabilize the forecasting. Based on the proposed the algorithm, three aspects' results are summarized, which includes the high accuracy of the very short term load forecasting, the gradually tuning property, and the results of whole day tuning. These results demonstrate the effectiveness of the proposed method and display a great improvement on the CMKs on the daily load forecasting.

Chapter 4

Optimal Planning and Dynamic Operation of Distributed Generation Method Based on Modified Multi-objective Optimization in Power Distribution System

4.1 Introduction

In recent years, the increased use of distributed energy, such as photovoltaic (PV), wind turbines and fuel cells, has gained wide acceptance because such sources of are renewable and environmentally friendly. However, due to misplacement, inappropriate sizing, and operation of DGs, they also pose a challenge to the distribution network operator because a high penetration of DG in certain places can result in voltage deviation and power losses [7, 122]. To tackle these problems, many methods have been advanced. In general, methods can be divided into three types [8]. The first type is the analytical method, such as [123] that proposes to use the power loss sensitivity factor to find the place and size of the DG. In [124], the Kalman filter algorithm is taken to find the DG size after researchers obtain the optimal location for the DG by analyzing power losses

in a steady state. The second type is the numerical method, which contains the gradient search and linear and nonlinear programming, etc. For example, [125] uses the mix integer nonlinear programming based integrated model to minimize the DG's investment and operation cost during the DG planning. The third type is the heuristic method, which includes genetic algorithm (GA), particle swarm optimization, Tabu search, and ant colony optimization, etc. Among these types, the third is robust and works well for large and complex DG planning problems [126]. Nondominated sorting genetic algorithm II (NSGA II), one of type three, is a powerful algorithm to solve multi-objective problems [127]. Rather than tackling the static case, this paper proposes the model based on modified NSGA II to plan and dynamically operate the DG with uncertainty loads and DG outputs, all the while minimizing the circuits' voltage deviation and power losses.

In this chapter, the multi-objective optimization and GA is first briefly introduced. The uncertainty of load generation and PV output are then modeled. The voltage sensitivity analysis is utilized to select the location for installation of the DG and the modified NSGA II is built by adding the fuzzy logic decision part to select the best solution. Second, to dynamically operate DG, the strategies to increase the method's computation speed are discussed. Finally, the method at IEEE 14 bus is tested, and it is compared with several optimal power flow methods: primal/dual interior point method (PDIPM) [128] and trust region based augmented Lagrangian method (TRALM) [128]. The results demonstrate the effectiveness of this method.

4.2 Introduction to Multi-objective Optimization and GA

As a decision-making problem, it is common to have more than one objectives, which usually conflict with each other. A multi-objective optimization (MOO) problem is an optimization problem that involves multiple objective functions [129]. Without losing the generality, the optimization problem can be considered as the minimization problem, since a maximization problem can be transformed to the minimization problem by multiplying a negative one. For a k objectives MOO problem, let $\mathbf{x} \in \mathbb{R}^n$, it can be defined as [130]:

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{Z}(\mathbf{x}) = (Z_1(\mathbf{x}), Z_2(\mathbf{x}), ..., Z_k(\mathbf{x})) \\ \text{subject to} : \ & g_i(\mathbf{x}) \leq 0, \qquad i \in [0, p] \\ & h_j(\mathbf{x}) = 0, \qquad j \in [1, q] \end{aligned} \tag{4.1}$$

where $k \geq 2$, p, q are the number of the inequality constrain and equality constrain. The $\mathbf{X}$ is defined as a *feasible decision space* if $\forall \mathbf{x} \in \mathbf{X}$ that satisfies the constrains in (4.1).

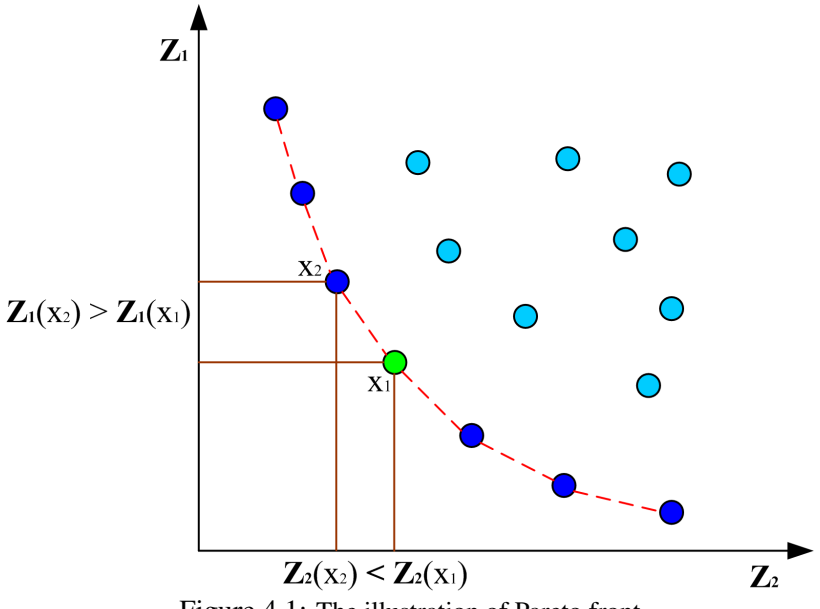

Figure 4.1: The illustration of Pareto front

In real applications, the MOO problem is nearly impossible to find a global solution for every objective function, and the solution that only optimizes one objective is not acceptable because it usually conflicts with other objectives [131]. Thus, it requires designing some criteria to measure the solutions. The commonly used concept is the *Pareto optimality*, which is defined: let $\mathbf{X}$ be the feasible decision space. For a point $\mathbf{x}^* \in \mathbf{X}$ is *Pareto optimal* if and only if there does not exist another point $\mathbf{x} \in \mathbf{X}$, such that $\mathbf{Z}(\mathbf{x}) \leq \mathbf{Z}(\mathbf{x}^*)$, and $\mathbf{z}(\mathbf{x}) < \mathbf{z}(\mathbf{x}^*)$ for at least one function [130]. The objective functions' values in the objective space, corresponding to the Pareto optimal set, are identified as the *Pareto front*. For example in figure 4.1, the dots symbolize of the feasible solutions of (4.1), and the dark blue and green dots are the Pareto front, which is connected by the

red dash line. Similarly, the vector of the objective function $\mathbf{Z}(\mathbf{x}^*)$ is nondominated if and only if there does not exist another vector $\mathbf{Z}(\mathbf{x})$, such that $\mathbf{Z}(\mathbf{x}) \leq \mathbf{Z}(\mathbf{x}^*)$ with at least one $\mathbf{z}(\mathbf{x}) < \mathbf{z}(\mathbf{x}^*)$. Otherwise, $\mathbf{Z}(\mathbf{x}^*)$ is dominated [130]. This concept of objective space nondomination is used in the evolutionary algorithm to search the solution for the MOO problem.

The genetic algorithms (GAs) are built based on the natural selection of evolutionary theory. The GAs simulate the evolution process in nature to select the solution for a single or multi-objective problems and are global optimization methods, which can converge to the global solution if provided enough computation resources [130].

Let $\mathbf{x} \in \mathbf{X}$ is a solution for the GAs. In the GAs terminology, then $\mathbf{x}$ is defined as a *chromosome*. The components in the $\mathbf{x}$ are defined as *genes*. The genes express the features of the chromosome, which encode solution space. To evolve the solution for the MOO problem, the GAs work on a collection of chromosomes, which is defined as *population*. To operate the population, two operators, *crossover* and *mutation*, are designed. Using the crossover operator, the GAs usually take two chromosomes, called *parents*, to formulate another new chromosome, called *offspring*. The fitness from the objective functions measures the quality of the offspring. With some mechanism, such as the *elitism*, the high-quality offspring would be selected in the next generation and so on, which contributes to the converge of the overall solution of MOO problem. With the mutation operator, the GAs usually work on the operation of the genes level. The mutation rate defines the degree of gene changing in the chromosome, with which the GAs avoid the local optima and introduce the diversity of the chromosomes [131]. For detailed introduction and explanation of MOO and GA, refer to [131] and [130].

Nondominated Sorting Genetic Algorithm II (NSGA II) is considered as one of the fast GAs. It has the advantage of low computation complexity, elitism, fast computation, and nondominated sorting [127]. In this work, the NSGA II is used to solve the MOO problem to reduce the active and reactive power loss and voltage deviations during the power distribution.

4.3 Problem Formation

4.3.1 Voltage sensitivity analysis for buses to install DG

Voltage stability is an important index in a power system. The installation of DG units in a distribution system could affect the voltage stability, which usually depends on DG's outputs and location [132]. To select the best place to install a DG, which would increase the voltage stability and reduce the voltage deviation, the voltage sensitivity analysis is taken in this work. In [133], the nonlinear differential algebraic equation is used to model the power systems, which can be linearized below:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_{P\theta} & J_{PV} \\ J_{Q\theta} & J_{QV} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} \tag{4.2}$$

Where the P represents the active power, Q represents the reactive power, θ represents the bus voltage angle, and V represents the bus voltage.

When assuming that the reactive load power is a constant and the term ΔQ equals zero, the following equation can be deduced from (4.2):

$$\Delta P = (J_{PV} - J_{P\theta} J_{Q\theta}^{-1} J_{QV}) \Delta V \tag{4.3}$$

$$\Delta V = (J_{RE})^{-1} \Delta P \tag{4.4}$$

where J_{RE} represents a reduced Jacobian matrix, which describes the relationship between the injected active power from the DG and the load bus voltage. This work models the load buses as PQ type and the DG with unity power factor, so from [133], (4.4) can be used to compute the voltage sensitivity profile when installing the DG. Otherwise, the VQ sensitivity needs to be considered.

4.3.2 Objective functions

The high penetration of DG has the potential to lead to the instability of voltage and increase the power losses [7]. When candidate buses to install the DG are defined, the operation of the DG is calculated based on the objective functions that target reducing the active and reactive power losses and the load voltage deviations for all buses.

The system's active and reactive power losses, P and Q respectively, are written as:

$$P_{Loss}^{ij} = P_i - P_j = \frac{(P_i^2 + Q_i^2)r_i}{V_i^2} \tag{4.5}$$

$$Q_{Loss}^{ij} = Q_i - Q_j = \frac{(P_i^2 + Q_i^2)x_i}{V_i^2} \tag{4.6}$$

The definitions of the first two objective functions are:

$$Min\big(\sum_{t=1}^{M}\sum_{j,j\neq i}^{N}\sum_{i}^{N} P_{Loss,t}^{ij}\big) \tag{4.7}$$

$$Min\big(\sum_{t=1}^{M}\sum_{j,j\neq i}^{N}\sum_{i}^{N} Q_{Loss,t}^{ij}\big) \tag{4.8}$$

Where i and j are the number of connected buses, N is the sum of buses in the circuit, M is the sum of time sample point, and define the resistance and reactance between the connected buses i and j as r_i and x_i respectively.

The third objective function is to minimize the load voltage deviation in the system:

$$Min\big(\sum_{t=1}^{M}\sum_{i}^{N}(V_{i,t} - V_{ref})\big) \tag{4.9}$$

In this work, the V_{ref} equals 1.

The constraints are the boundary of the reference buses' voltage magnitude and angles, the boundary of the generators' real and reactive power injections, the boundary of tap ratio [134], and the boundary of DG output.

4.4 Modeling And Simulation

In this work, the study case is the IEEE 14 bus. To make the simulation dynamically, the load data and DG (Photovaltaic panels) outputs are generated in 15-minute intervals for every bus within the boundary for a day (i.e. 96 groups in a day). Based on the load and DG output, the voltage sensitivity is analyzed to select the places for installing the Photovaltaic (PV) panels. Finally, the modified NSGA II model is used to dynamically calculate the optimal operation of the DG.

4.4.1 System under study

For the study case, the IEEE 14 bus case has two generators, located at bus 1 and bus 2. Three synchronous compensators are placed at bus 3, 6 and 8. The IEEE case also includes two transformers and eleven PQ buses, which have total loads as 259 MW and 73.5 MVAR. The system's data is given on $100MVA$ base. The maximum and minimum limits of voltage magnitude is $0.94p.u.$ and $1.06p.u.$. The system is depicted in figure 4.2 [135].

4.4.2 Uncertainty modeling and data generation

To model the uncertainty of loads, a one-day shape of the loads from the local electrical utility company is employed (see figures 4.3 and **Algorithm 1**. for details). According to its shape, the sum of loads for every 15 minutes is allocated, which is bounded by the default case of IEEE 14 buses, i.e. the maximum sum of loads is considered as the default sum of loads in the IEEE 14 bus. The loads are then randomly generated for each bus according to the loads that is allocated, as seen in figure 4.3. In this way, the generated loads may contain reasonable random uncertainty for every bus, while keeping the same load shape as the real case from the local utility.

In figure 4.3, the blue dash line is the total loads for a day. The green solid line is the bus 8 loads in a day, and the red dash lines with stars in their terminals are the load ranges for all the 11 PQ buses of IEEE 14 buses. In order to explain the process, the steps are are summarized in

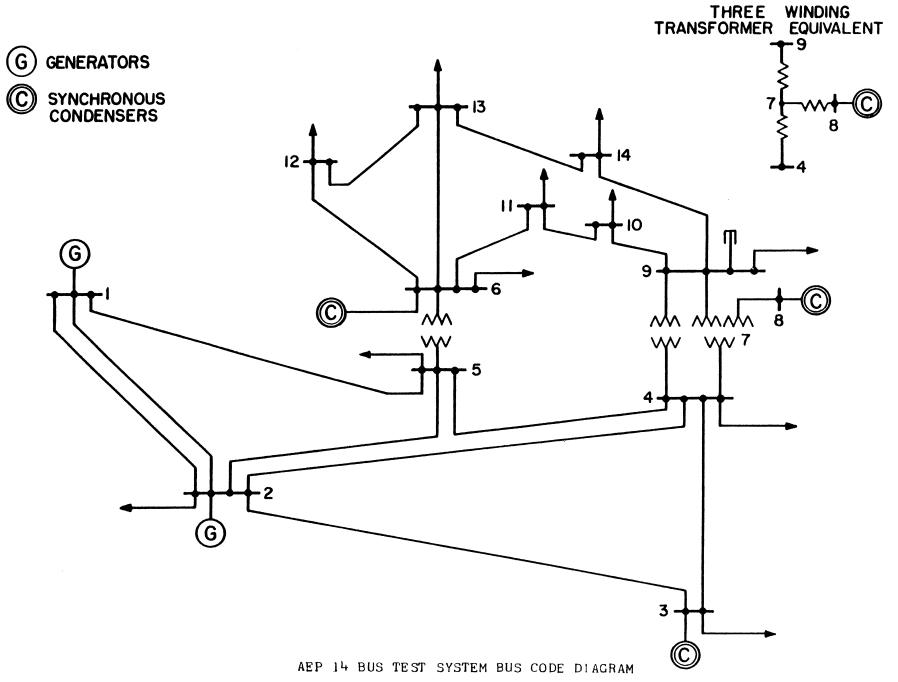

Figure 4.2: The IEEE 14 bus system.

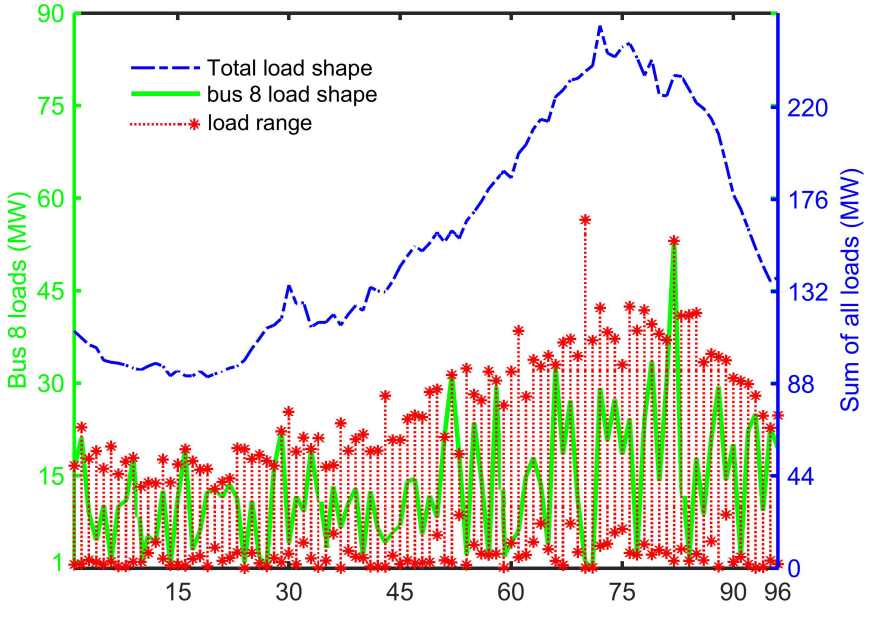

Figure 4.3: Randomly generated loads and total loads.

Algorithm 1.

Algorithm 1 generate the daily random load for buses

1: **Input**: $p_{act} = 259$, $p_{react} = 73.5$, $\mathbf{l}_{orig}$ = one summer daily historical load from the local utility with 15-min intervals
2: **Output**: $\mathbf{r}_{bus}$, $\mathbf{q}_{bus}$ ▷ Generated daily active and reactive load data for each bus
3: $\mathbf{r}_l = \frac{\mathbf{l}_{orig}}{\max(\mathbf{l}_{orig})}$ ▷ Get the daily load curve ratio for each time interval
4: $\mathbf{p}_{dly} = p_{act}\mathbf{r}_l$, $\mathbf{q}_{dly} = q_{react}\mathbf{r}_l$ ▷ Get the 15-min interval active and reactive load for all buses
5: **for** $tmInter \leftarrow 1, 96$ **do**
6: $\mathbf{g}_p = rand(11, 1)$, $\mathbf{g}_q = -5 + 10rand(11, 1)$
7: $\mathbf{r}_p = \frac{\mathbf{g}_p}{sum(\mathbf{g}_p)}$, $\mathbf{r}_q = \frac{\mathbf{g}_q}{sum(\mathbf{g}_q)}$ ▷ Generate the random ratio of load for each buses
8: $\mathbf{r}_{bus}(1:11, tmInter) = \mathbf{p}_{dly}(tmInter)\mathbf{r}_p$ ▷ Generate the random load for 11 PQ buses
9: $\mathbf{q}_{bus}(1:11, tmInter) = \mathbf{q}_{dly}(tmInter)\mathbf{r}_q$ ▷ Generate the random load for 11 PQ buses
10: **end for**

Algorithm 2 generate the daily random PV output

1: **Input**: $tm_{cld} = 3$, $cld_{bd} = 360$ (i.e.: 3 hours), $\mathbf{g}_{pv}$ = idea PV output curve from local utility (half-min interval data)
2: **Output**: $\mathbf{u}_{pv}$ ▷ Generated the daily PV output
3: Initialization: $\mathbf{u}_{pv} = \text{zeros}(2880, 1), nonZr = \text{find}(\mathbf{g}_{pv} > 0)$
4: $pvStt = nonZr(1)$, $pvStp = nonZr(end)$ ▷ Get when the PV generates power
5: $tmSun = pvStp - pvStt + 1$ ▷ time length of the PV output
6: **for** $i \leftarrow 1, tm_{cld}$ **do**
7: $cldInd = \text{floor}(\text{rand}(1)tmSun)$ ▷ Generate when would have clouds
8: $cldLen = \text{floor}(\text{rand}(1)cld_{bd})$ ▷ Generate the clouds lasting time
9: $\mathbf{l}_{cld} = \text{rand}(1, cldLen)$ ▷ Generate random ratio for PV output
10: $chgeStt = cldInd + pvStt$
11: $\mathbf{u}_{pv}(chgeStt : chgeStt + cldlen - 1) = \mathbf{g}_{pv}(chgeStt : chgeStt + cldLen - 1) \circ \mathbf{l}_{cld}$ ▷ The symbol $\circ$ means the Hadamard product operation for vector
12: **end for**

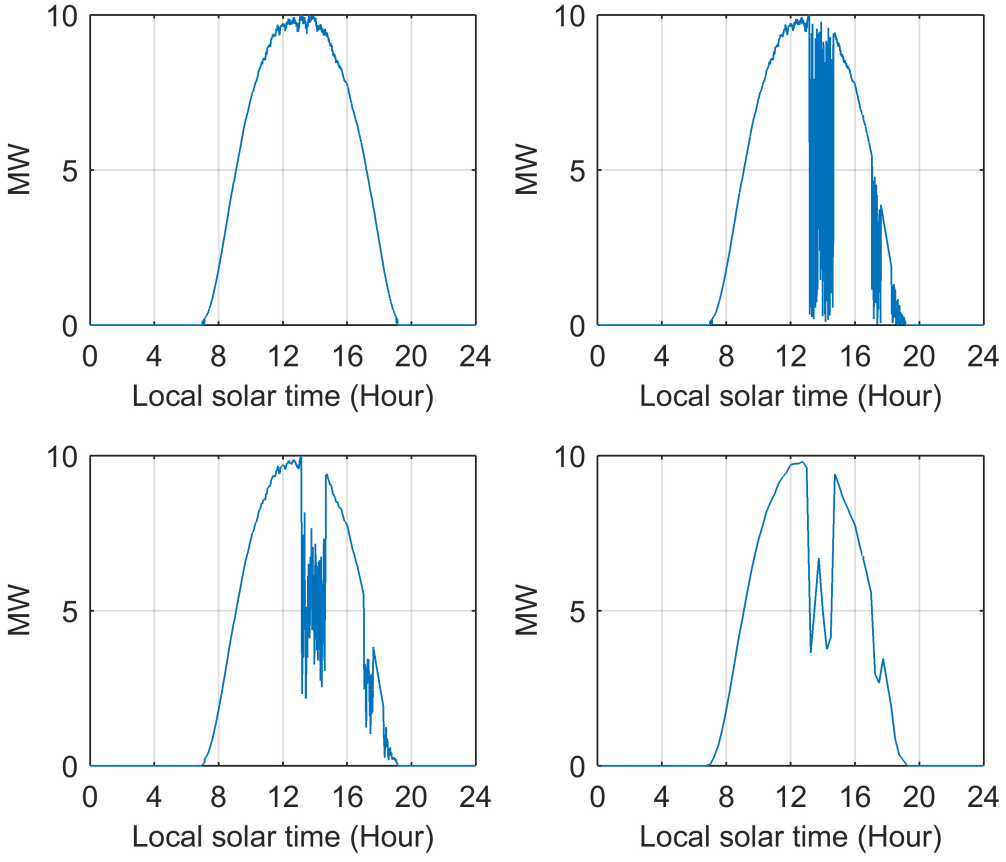

Figure 4.4: PV panels output for uncertainty modeling.

To model the uncertainty of the PV panels output, cloudy index, times, and lasting length are added in the ideal summer PV output curve, which is obtained from the local electrical utility company. The generation steps are summarized in **Algorithm 2** with randomized parameters as cloudy index, cloud lasting length, and the PV output ratio. The generated PV outputs are displayed in figure 4.4. The shapes of the PV output are the same for the local test case. The output value marked here is for IEEE 14 case.

In figure 4.4, the upper left figure is the ideal PV output curve. The upper right figure is the output using the data generated by **Algorithm 2**. The lower left figure illustrates the result when the moving average is applied to filter the noise and the lower right figure is the final output used in the model.

4.4.3 Voltage sensitivity analysis to define the candidate buses

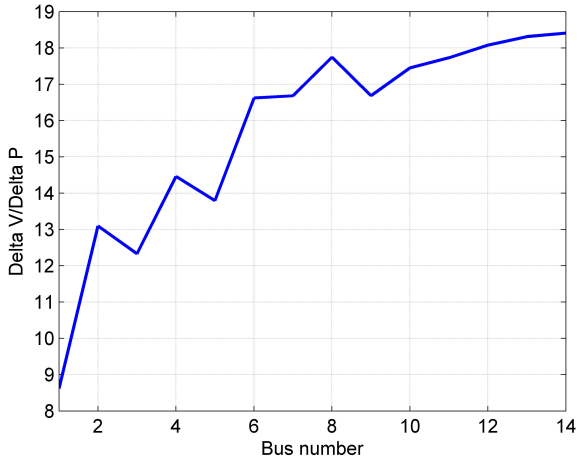

Figure 4.5: The voltage sensitivity for each bus in a day.

Using (4.4), the voltage sensitivity of each bus is computed for every time interval. For the IEEE 14 bus test case, there are 14 buses and 96 time intervals in a day. Thus, for each bus, the voltage sensitivity is calculated with the load that is generated in part 4.4.2 for 96 time intervals. The sensitivities for each bus are summed up and plotted it in figure 4.5. Figure 4.5 shows buses 12, 13 and 14 are the most sensitive that could be the candidate places to install the DG.

4.4.4 Modified NSGA II model for DG operation

After the DG installation location is defined, determine the optimal size and operation of the DG to satisfy several objectives is the next target. For multiobjective problems, which has objective functions conflict with each other, finding multiple compromised solutions is more reasonable than a single best solution, since a solution may not exist. The NSGA II is currently one of the most popular algorithms that target multiobjective problems, which is utilized here. The steps that are used and modified in this chapter are introduced.

(1) The parent population, T, are randomly generated within the boundary defined in the problem formulation. The chromosome of each individual is composed by the buses' voltage magni-

tudes and angles, generators' active and reactive power outputs, tap ratios, shunt reactance, and the DG output for a specific bus calculated from the voltage sensitivity analysis.

(2) Nondominated sorting is performed to classify the population. The binary tournament selection is then used to get winners based on the level of nondomination and the crowding distance between the competitors. Intermediate crossover and Gaussian mutation are taken to generate the next generation from previous winners. Here, the mutation rate is set to 0.04. The generated offspring and their parents are saved and sorted based on nondomination. If the population is less than T, then the remaining part of population will be randomly generated. The new generation is produced and the algorithm is ended if the stop criteria are satisfied, e.g. it achieves maximum number of generation: G. Here, T equals 480, and G equals 36. Otherwise, it will repeat the previous steps.

(3) A fuzzy logic decision model is built. When the multiobjective problems are solved, a Pareto front is obtained. However, in the actual application, only a "best" solution is needed to solve the MOO problem. To find the best compromised solution from the Pareto front, the fuzzy logic technology is used [136], which can obtain a good compromised individual from the last population in last generation that is generated. Take the figure 4.1 for example, this fuzzy logic decision model tries to find the *green dot* (i.e.:$\mathbf{x}_1$) as the compromised the solution because it has decent values for both objective functions $Z_1(\mathbf{x}_1), Z_2(\mathbf{x}_1)$. The model's inputs are active power losses, reactive power losses, and voltage deviations. The model outputs a score for each individual, and the one with the highest score is selected. Their membership functions are a general Z-shaped function, a GAUSS function, and a S-shaped function, respectively, but with different ranges and various parameters. For example, if the model emphasizes the importance of the minimization of the active power losses, the C parameter of the GAUSS function in the active power loss input approaches the 'good' side and it also has more weight in the fuzzy rules. A figure 4.6 shown a detail. This model is written in the scripts that is incorporated in the optimization algorithm.

To find the optimal operation of the DG for one time interval, the modified NSGA II method is built up based on previous steps, since the DG's energy output to grid is one of components inside

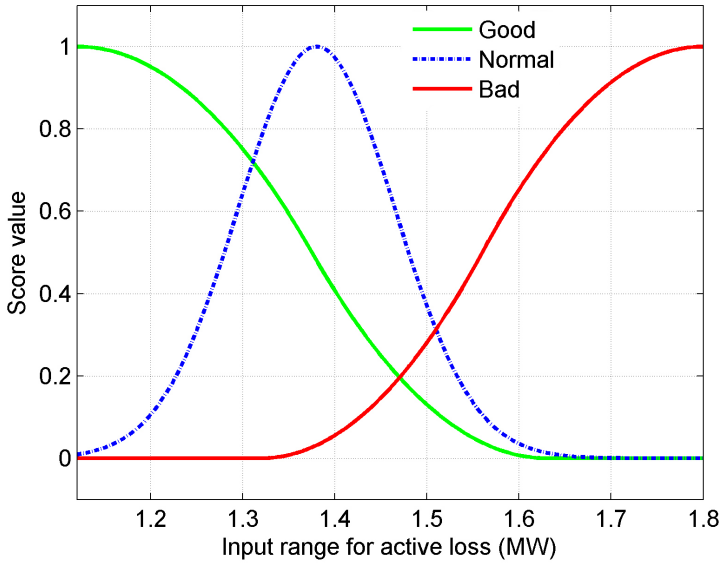

Figure 4.6: The membership functions for active power losses.

of the chromosome. Simultaneously, this model provides good parameters for other equipment in the power distribution, such as the generators' outputs and transformers' tap ratios.

4.4.5 The dynamic DG operation based on modified NSGA II

To make the modified NSGA II model dynamical, the loads data is incorporated with PV outputs that are generated in Part 4.4.2. The modified NSGA II model is then continuously employed to solve the multiobjective problem with the loads and the PV outputs changed sequentially. Meanwhile, the optimal size of DG is obtained during this period. The issues in applying the modified NSGA II dynamically depends mainly on the following [137]: assume the model spends a time t_r to run one generation and there are G generations for certain loads. Suppose the loads are not changed within a time interval T_l and $Gt_r < T_l$. Thus, if T_l is too small, it may not get the best Pareto front with a limited number of generations; if T_l is too big, the loads may change to a large degree, which requires more generations to get the best Pareto front. Therefore, increasing the model's computation speed for each generation is a possible solution for the issues.

Next, we try to reduce the computation time for the modified NSGA II model when it tracks the optimal Pareto front during the simulation.

1) The initialization space is reduced. The first step of the NSGA II is randomly generating the population inside of initialization space. If the space can be reduced, then less number of populations are needed to distribute over the space, which means less time is used to evaluate them. This can be done because based on an assumption that when the loads are selected to be the extreme case, to get the optimal Pareto front, the variables in the chromosome also range to an extreme degree. When the loads are in a decent case, then the variables also range to a comparatively small degree. If the short-term and very short-term load forecasting is accurate, as in Chapter 3, this initialization space may be compressed further, because the NSGA II can be employed in advance based on the accurate load forecasting, and the range of variables initialization space can be calculated and confined in advance.

2) The last population from the last generation is incorporated into the next generation's population. If loads in the time interval T_l do not change too much, or if the T_l is small, the parameters of the NSGA II may be similar to the previous time interval to get the optimal frontier. In this case, the population inherited from the last generation will help the model to find the new optimal Pareto front quickly. To maintain the diversity of the new population, the mutation rate is increased to 0.09. Also, the best percentage of a population that the current population inherited from the last generation still needs to be tested.

The general procedures for our dynamic operation are summarized in figure 4.7. With previous strategies, the operation time of the modified NSGA II model is reduced to half of the original case, and the optimization performances are still good. By taking the maximum production of DG during this period, the optimal size of DG can be determined. This dynamic operation of DG, based on the modified NSGA II model, not only provides us a way to operate DG optimally, but also guides us to operate other electrical components effectively. The results are presented in the result part.

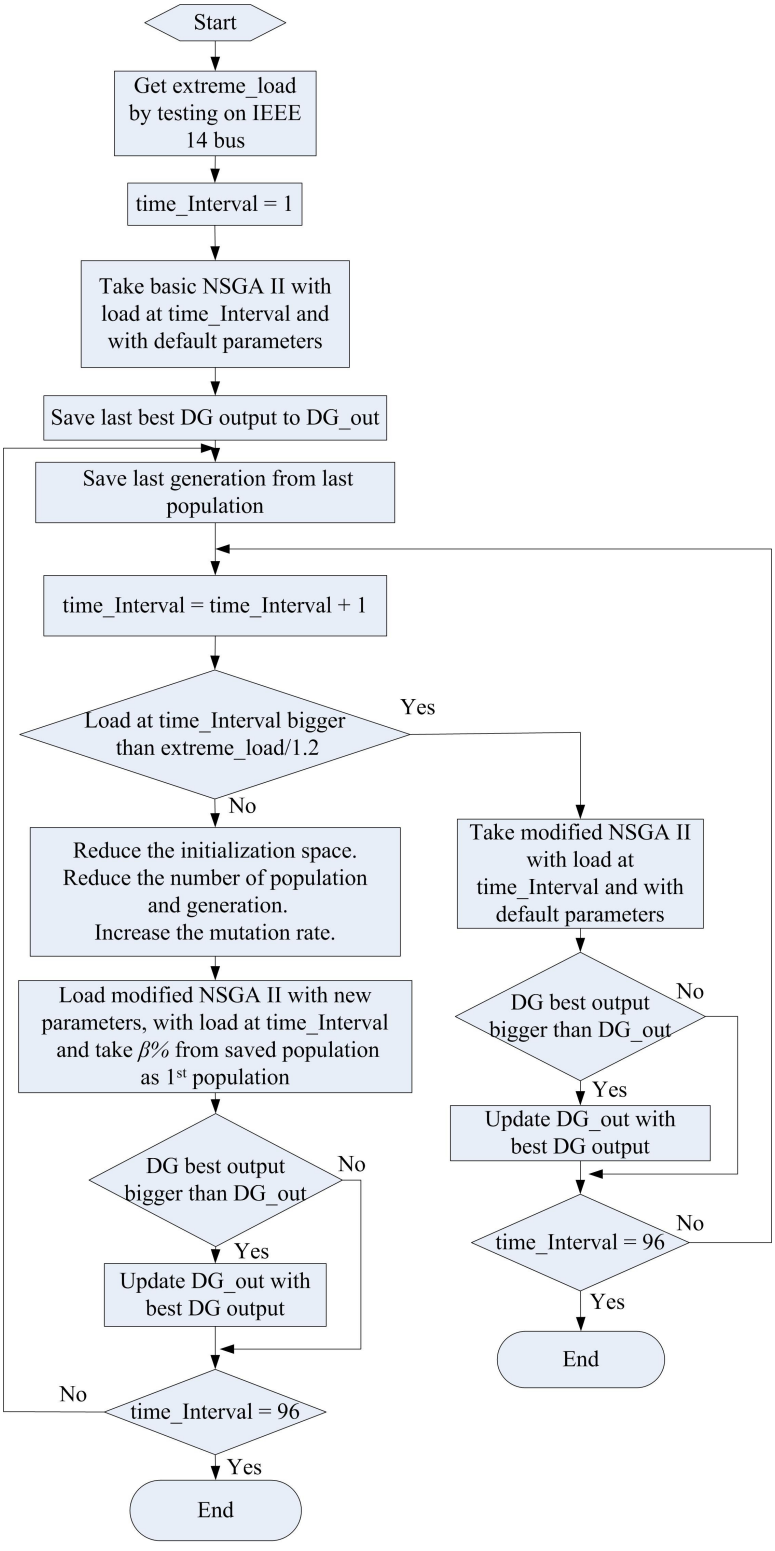

Figure 4.7: The flow chart of the dynamic operation of a DG.

4.5 Results

Many tests have been taken to verify the method at the IEEE 14 bus. The first case compares the modified dynamic NSGA II with the case without optimization, and the optimization methods PDIPM and TRALM. The second compares the modified dynamic NSGA II with traditional NSGA II.

Case 1: comparing the modified dynamic NSGA II with the optimal methods PDIPM, the TRALM, and no optimization case.

Here, the maximum DG output is 10 MW, where DG is considered as the PV panels. We also assume there is an energy storage system that can store the extra power from the DG that is not injected into the grid. The generation is set as 25, and the population is selected as 400 for modified dynamic NSGA II model. According to the voltage sensitivity analysis, the bus 12, 13 and 14 are the best places to install the DG. Here, the DG installation tests are extended to include bus 7 to bus 11.

To calculate the daily average power losses and voltage deviation at each time point, the following equations and steps are used: assume the PV is installed at i^{th} bus, $i \in [7, 14]$ at t^{th} time point $t \in [1, 96]$, the generated the loads and PV outputs are used to compute the power losses and voltage deviations. The active power losses: $actP^i_{loss}(t)$, the reactive power losses $reactP^i_{loss}(t)$ and voltage deviation $volDev^i(t)$ are computed by each methods. The daily average active power losses at i^{th} bus is computed by: $\overline{dlyActP^i_{loss}} = \frac{1}{96}\sum_{t=1}^{96} actP^i_{loss}(t)$. The daily average reactive power losses at i^{th} bus is computed by: $\overline{dlyReactP^i_{loss}} = \frac{1}{96}\sum_{t=1}^{96} reactP^i_{loss}(t)$. The daily average voltage deviation $\overline{dlyVolDev}_i = \frac{1}{96}\sum_{t=1}^{96} volDev^i(t)$. The $\overline{dlyActP^i_{loss}}$, $\overline{dlyReactP^i_{loss}}$, and $\overline{dlyVolDev}_i$ are plotted for every method in figure 4.8.

In figure 4.8, the black bars represent the results without any optimization on the active and reactive power losses and voltage deviation when installing a DG in the corresponding bus. The blue bus tells the results generated by the PDIPM, and the cyan bars displays the results computed by the TRALM. The yellow, red and green bars are active and reactive power loss and voltage

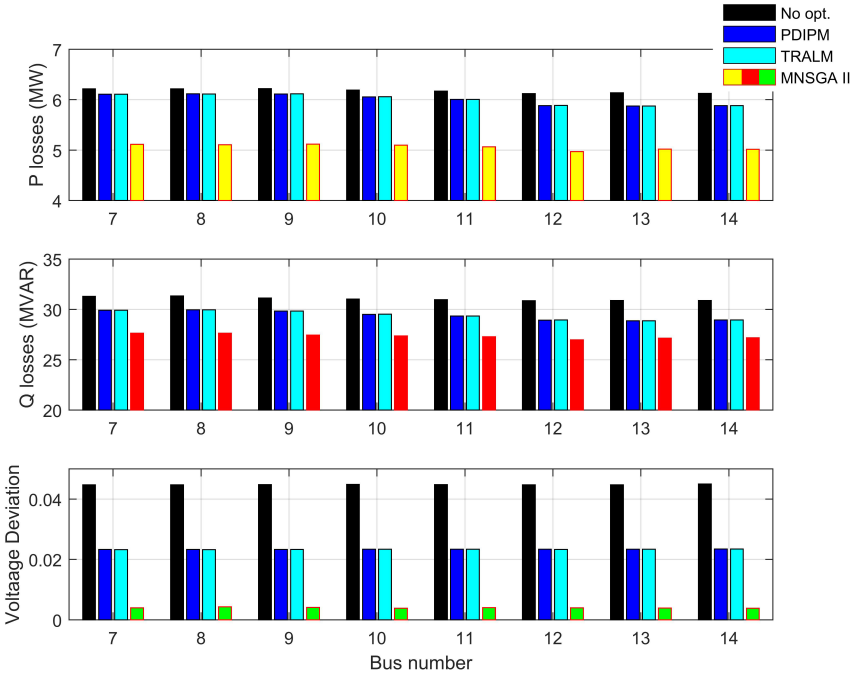

Figure 4.8: The daily average power and voltage deviation comparison from modified NSGA II, the PDIPM, the TRALM, and no optimization.

deviation with modified NSGA II method. It is clearly to see that using the modified NSGA II method will save lots of energy and also keep the circuit more stable. Meanwhile, the results from the PDIPM and the TRALM methods are almost the same, but both of them are better than the results from the method without optimization.

To calculate the daily energy losses, the equation $W = Pt$ is used. Since the data in this system is sampled by 15-min, the energy loss for i^{th} bus at t^{th} time point is: $W^i_{loss}(t) = 0.25 act P^i_{loss}(t)$. The daily energy losses for each bus is computed as: $dlyW^i_{loss} = \sum_{t=1}^{96} W^i_{loss}(t)$. To calculate the percentage of the energy losses, the $perEneg^i_{loss} = \frac{dlyW^i_{loss}}{dlyWall} \times 100\%$, where the $dlyWall$ is the sum of energy that is generated by the generator during a whole day. The energy loss and percentage of energy loss for each method are plotted in figure 4.9. The black bars represent the case without any optimization, the blue bars represent the PDIPM method, and the cyan bars represent the TRALM method. The results from the modified NSGA II is plotted in yellow bar and green bar.

The Modified NSGA II has the minimum energy losses when compared with other methods,

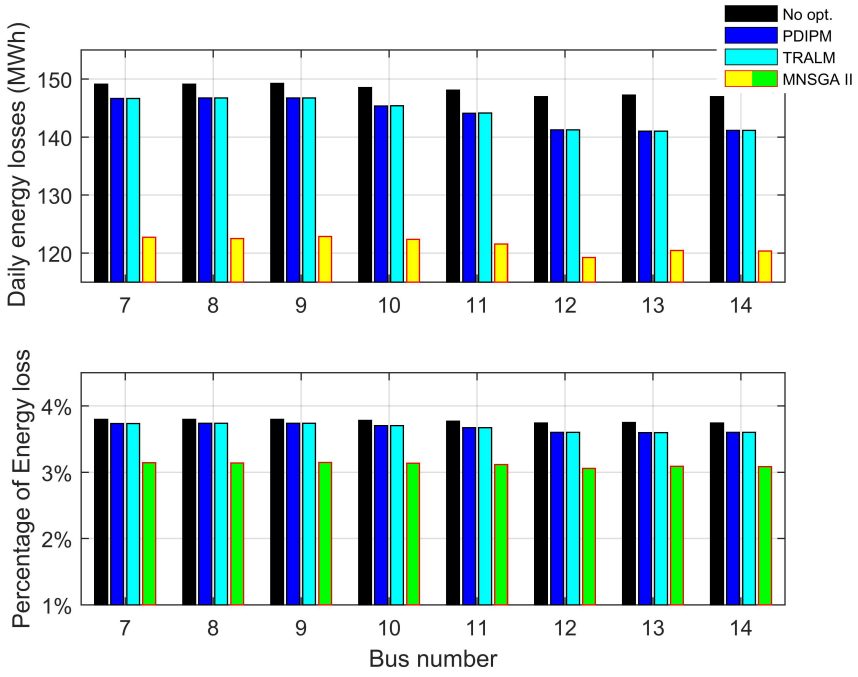

Figure 4.9: The daily energy losses from modified NSGA II, the PDIPM, the TRALM, and no optimization.

as the percentage of the energy losses at approximately 3%. The energy losses for the PDIPM and the TRALM methods are better than the case without optimization.

Moreover, when the PV is installed on bus 12, bus 13 or bus 14, the power and energy loss are smaller than when the PV is installed on other buses. This feature is consistent with all the methods of comparing, and also compatible with the voltage sensitivity analysis that the bus 12, 13 and 14 are the good candidates to place the PV.

Case 2: the comparison of the basic NSGA II model and the modified dynamic NSGA II model

The generation is set as 36, and the population is set as 480 for the basic NSGA II model. The parameters for the modified NSGA II model are set the same as in *case 1*.

The same steps and equations with **Case 1** are used to calculate the average power losses, voltage deviation, and the percentage of the energy losses here. In both figure 4.10 and figure 4.11, the blue bars represent results from the traditional NSGA II. The yellow, green, and red bars represent the results from the modified NSGA II. From figure 4.10, the modified NSGA II

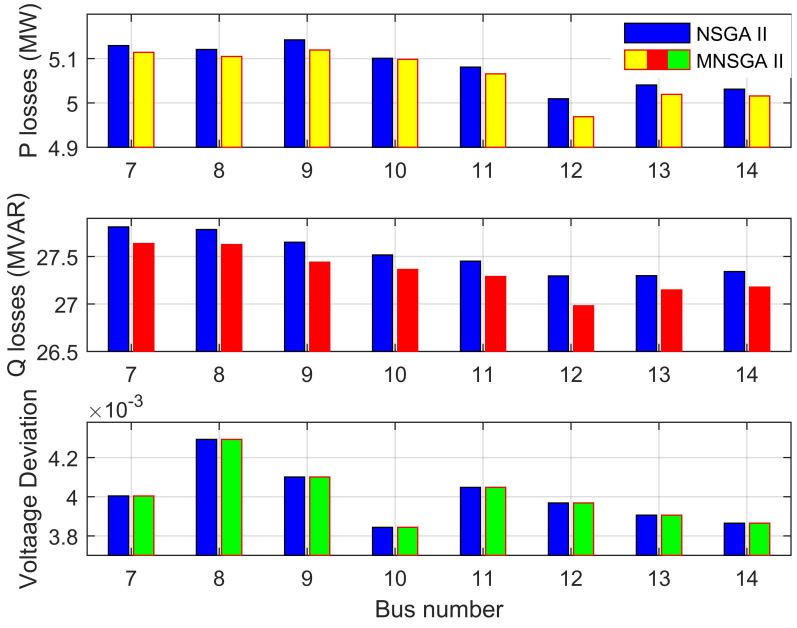

Figure 4.10: Comparison of power losses with NSGA II and modified NSGA II.

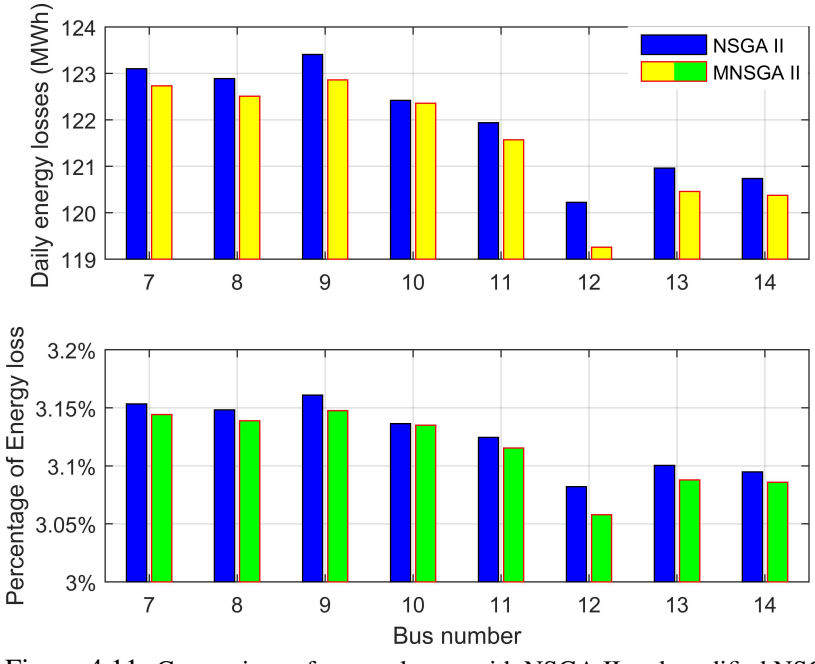

Figure 4.11: Comparison of energy losses with NSGA II and modified NSGA II.

is better than the NSGA II at the power loss, and the modified NSGA II has almost the same voltage deviation with the NSGA II. From figure 4.11, the modified NSGA II is also better than the traditional NSGA II from both the energy losses and the percentage of the energy losses.

There are two important benefits from the modified NSGA II in this work. One is that it runs much faster. With modifications, to train one generation, it spends 70.575 seconds instead of 133.612 seconds, which means it is almost twice as fast as the case without modifications. The other is it has fewer power losses, and with the voltage deviation is almost the same with the traditional NSGA II. The cases were tested on a computer with 8 GB installed memory, 4 Cores' CPU and each processor is 2.00GHZ.

4.6 Conclusion

In this chapter, a method to plan and dynamically operate DG is proposed, which is based on modifications of the NSGA II. To test the method, the uncertainty of the load and the PV output are considered, and the algorithms are designed to incorporate the uncertainty to generate load and PV outputs. The voltage sensitivity is applied to select the candidate buses for installing the DG. The NSGA II algorithm is modified by adding the fuzzy logic decision model so to choose the "best" solution from the Pareto front. Two strategies are designed to increase the modified dynamic NSGA II computation speed. The modified dynamic NSGA II then used in the daily operation of test cases, i.e. the IEEE 14-bus, displays great ability in increasing the computation efficiency and reducing the power losses and voltage deviations.

Chapter 5

Summary and Future Research Directions

In this dissertation, we design a framework to optimally operate power distribution with distributed generators based on the machine learning techniques. The contributions are summarized, and then the potential future research is addressed.

5.1 Summary of This Dissertation

In Chapter 2, a comparative study for the peak load forecasting at distribution feeder level circuits are investigated. First, two years of private load data from the local utility and two years of public load data from Texas utility are analyzed. The correlation analysis is applied between peak load and weather factors. Next, a new nonparametric, Bayesian Additive Regression Trees (BART), is introduced to do peak load forecasting. Since the BART method takes a amount of time to generate the forecasting, the composite kernel methods based on Gaussian Process Regression (CKGPR) are designed. Then these methods have been compared with Multiple Linear Regression (MLR) method and the Support Vector Regression (SVR) method based on the private residential area and public business area load data. Thorough comparison results are presented based on five forecasting measurements. The BART has the best forecasting accuracy among all the indices, and the CKGPR also has counterpart forecasting results but with less computation time. Meanwhile,

the forecasting accuracy difference between two areas is analyzed. Last, influential weather and human factors are summarized.

In Chapter 3, a new framework of distribution-level time series short-term and very short-term load forecasting is proposed. This framework may be divided into two parts. In the first part, the composite Matérn kernels of Gaussian process regression is designed to do day-ahead load forecasting based on the thorough analysis of two areas' load data and kernels comparison. A data selection algorithm is also designed to improve the prediction accuracy of the composite kernels further. In the second part, a daily curve tuning algorithm (CTA) is designed based on creatively using the dictionary learning algorithm, K-SVD, to further improve the forecasting result from the first part. Three steps are summarized for the CTA. In step one, the new dictionary is built by using the K-SVD to decompose the output of the composite Matérn kernels based on the historical load data. In step two, for a certain length of atoms (T0), the tuned curves are generated by using the K-SVD to learn the known daily load and curve selection model is then designed to select the best-tuned curve based on the linear regression models with forecasting errors as feedback. In the last step, the final tuned curve is selected by the minimization of the mean daily load difference for different length of atoms. Three aspects results, very short-term forecasting, gradually tuning property and whole day tuning result, are summarized based on the 15-min interval, two-year private data from the residential area of Albuquerque and two-year public data from the business area at the north central Texas. The results demonstrate the high effectiveness of this framework for the distribution-level of circuits.

In Chapter 4, we proposed an optimal method to plan and dynamically operate the distributed generation (DG) based on the modified nondominated sorting genetic algorithm II (NSGA-II). First, the uncertainty of load and DG (photovoltaic panels) output are considered. Second, the placement of a DG is defined by a voltage sensitivity analysis. To find the optimal daily operation of a DG, a multiobjective problem is formulated that focuses on the minimization of a circuits voltage deviations, active and reactive power losses. To solve the problem, the traditional NSGA II is modified by incorporating a fuzzy logic decision model. The fuzzy logic model selects an optimally compromised solution from the Pareto front by analyzing its weights of voltage devia-

tions, active and reactive power losses. Furthermore, to operate a DG optimally and dynamically, the methods computation speed is crucial. To increase the modified NSGA II computation speed, the population initialization space is reduced, and the population is selected and saved for the next generation based on load analysis and experiments. The method is tested on the IEEE 14 bus and a local residential circuit. The results on reducing the power losses, voltage deviations, and increasing the algorithm speed demonstrate the effectiveness of this method.

5.2 Future Research Directions

In Chapter 2, the possible extension will include exploiting the explanatory abilities of the BART algorithm. The selection of the prior is essential to optimize results and needs further study. In particular, attention should be paid to how inference can be applied to adjust the prior variance. The number of trees is a free parameter which must be carefully validated to achieve optimum performance. Regarding the data, this Bayesian method is promising and can show improvement in performance in contrast to other regression methods if additional information is applied, such as the solar radiation and demographic data.

In Chapter 3, it is worth to point out that the spectral mixture kernel [138] may be a good candidate for the day-ahead load forecasting, instead of using the CMKs, because the spectral mixture kernel can approximate any stationary covariance kernel to a arbitrary precision [138]. In the CTA, the linear dictionary K-SVD plays an important role to generate the dictionaries and their coefficients. It is nice to point out that other linear dictionary learning algorithm, such as the recursive least squares dictionary learning algorithm [119], may have a counterpart effect in the CTA framework. Moreover, the predicted mean value of the daily load, generated and adjusted by the regression models in real-time, may be predicted day-ahead by regression model in Chapter 2 if its correlated factors or prior knowledge could be seized.

In Chapter 4, the modified NSGA-II framework may be extended to test with large circuits of more than one thousand buses. In real applications, a fast response to the load demand is crucial to

the power network. The evolutionary method needs to spend a time to derive the optimal solutions, so it is important to design a mechanism to improve the efficiency of the algorithm when applied to a large circuit. One possible solution is running the algorithm in advance with accurate short-term and very short-term load forecasting. Then, the derived optimal parameters may be used after that.

References

[1] Kroposki, Benjamin, Pankaj K. Sen, and Keith Malmedal. "Optimum Sizing and Placement of Distributed and Renewable Energy Sources in Electric Power Distribution Systems." *IEEE Transactions on Industry Applications*, 49.6 (2013): 2741-2752.

[2] Ackermann, T., Andersson, G., and Sder, L. (2001). "Distributed generation: a definition." *Electric power systems research*, 57(3), 195-204.

[3] Daly, Peter A., and Jay Morrison. "Understanding the potential benefits of distributed generation on power delivery systems." *In Rural Electric Power Conference, IEEE*, 2001, pp. A2-1. 2001.

[4] Pepermans, G., Driesen, J., Haeseldonckx, D., Belmans, R., and Dhaeseleer, W. (2005). "Distributed generation: definition, benefits and issues." *Energy policy*, 33(6), 787-798.

[5] El-Khattam, W., and Salama, M. M. A. (2004). "Distributed generation technologies, definitions and benefits." *Electric power systems research*, 71(2), 119-128.

[6] Atwa, Yasser Moustafa, and E. F. El-Saadany. "Optimal allocation of ESS in distribution systems with a high penetration of wind energy." *IEEE Transactions on Power Systems* 25.4 (2010): 1815-1822.

[7] Arya, L. D., Atul Koshti, and S. C. Choube. "Distributed generation planning using differential evolution accounting voltage stability consideration." *International Journal of Electrical Power and Energy Systems* 42.1 (2012): 196-207.

[8] Keane, Andrew, et al. "State-of-the-art techniques and challenges ahead for distributed generation planning and optimization." *IEEE Transactions on Power Systems* 28.2 (2013): 1493-1502.

[9] D. Q. Hung, N. Mithulananthan, and R. C. Bansal, "Analytical expressions for DG allocation in primary distribution networks," *IEEE Trans. Energy Convers.*, vol. 25, no. 3, pp. 814820, Sep. 2010.

[10] B. Banerjee and S. M. Islam, "Reliability based optimum location of distributed generation," *Int. J. Electr. Power Energy Syst.*, vol. 33, no. 8, pp. 14701478, Oct. 2011.

[11] F. S. Abu-Mouti and M. E. El-Hawary, "Heuristic curve-fitted technique for distributed generation optimisation in radial distribution feeder systems," *IET Gener., Transm., Distrib.*, vol. 5, no. 2, pp. 172180, Feb. 2011.

[12] M. Gomez-Gonzalez, A. Lpez, and F. Jurado, "Optimization of distributed generation systems using a new discrete PSO and OPF," *Elect. Power Syst. Res.*, vol. 84, no. 1, pp. 174180, Mar. 2012.

[13] S. Ghosh, S. P. Ghoshal, and S. Ghosh, "Optimal sizing and placement of distributed generation in a network system," *Int. J. Electr. Power Energy Syst*., vol. 32, no. 8, pp. 849856, Oct. 2010.

[14] M. H. Moradi and M. Abedini, "A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems,"*Int. J. Electr. Power Energy Syst*., vol. 34, no. 1, pp. 6674, Jan. 2012.

[15] M. F. Akorede, H. Hizam, I. Aris, and M. Z. A. Ab Kadir, "Effective method for optimal allocation of distributed generation units in meshed electric power systems," *IET Gener., Transm., Distrib.*, vol. 5, no. 2, pp. 276287, Feb. 2011.

[16] F. Rotaru, G. Chicco, G. Grigoras, and G. Cartina, "Two-stage distributed generation optimal sizing with clustering-based node selection," *Int. J. Electr. Power Energy Syst*., vol. 40, no. 1, pp. 120129, Sep. 2012.

[17] Georgilakis, Pavlos S., and Nikos D. Hatziargyriou. "Optimal distributed generation placement in power distribution networks: Models, methods, and future research." *IEEE Trans. Power Syst* 28.3 (2013): 3420-3428.

[18] Ehsan, Naderi, Hossein Seifi, and Mohammad S. Sepasian. "A Dynamic Approach for Distribution System Planning Considering Distributed Generation." *IEEE Trans. Power Delivery* 27.3 (2012): 1313-1322.

[19] Soroudi, Alireza, and Mozhgan Afrasiab. "Binary PSO-based dynamic multi-objective model for distributed generation planning under uncertainty." *IET renewable power generation* 6.2 (2012): 67-78.

[20] Chan, Shing-Chow, et al. "Load/price forecasting and managing demand response for smart grids: Methodologies and challenges." *Signal Processing Magazine, IEEE* 29.5 (2012): 68-85.

[21] Hernandez, L., Baladron, C., Aguiar, J. M., Carro, B., Sanchez-Esguevillas, A. J., Lloret, J., and Massana, J. (2014). "A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings." *IEEE Communications Surveys and Tutorials*, 16(3), 1460-1495.

[22] A. Papalexopoulos and T. Hesterberg, "A regression-based approach to short-term system load forecasting," *IEEE Trans. Power Syst.*, vol.5, no. 4, pp. 15351547, Nov. 1990.

[23] Hong, Tao, Pu Wang, and H. Lee Willis. "A Nave multiple linear regression benchmark for short term load forecasting." *Power and Energy Society General Meeting, 2011 IEEE*. IEEE, 2011.

[24] Douglas, Andrew P., et al. "The impacts of temperature forecast uncertainty on Bayesian load forecasting." *IEEE Transactions on Power Systems* 13.4 (1998): 1507-1513.

[25] H. T. Yang and C. M. Huang, "A new short-term load forecasting approach using self-organizing fuzzy ARMAX models," *IEEE Trans. Power Systems*, vol. 13, no. 1, pp. 217225, 1998.

[26] Cho MY, Hwang JC, Chen CS. "Customer short-term load forecasting by using ARIMA transfer function model." In: *Proceedings of the international conference on energy manage power delivery*, vol. 1. 1995. p. 31722.

[27] Bowden, Nicholas, and James E. Payne. "Short term forecasting of electricity prices for MISO hubs: Evidence from ARIMA-EGARCH models." *Energy Economics* 30.6 (2008): 3186-3197.

[28] Nie, Hongzhan, et al. "Hybrid of ARIMA and SVMs for short-term load forecasting." *Energy Procedia* 16 (2012): 1455-1460.

[29] J. H. Park, Y. M. Park, and K. Y. Lee, "Composite modeling for adaptive short-term load forecasting," *IEEE Trans. Power Systems*, vol. 6, no. 2, pp. 450457, 1991.

[30] S. Sargunaraj, D. P. Sen Gupta, and S. Devi, "Short-term load forecasting for demand side management," *IEE Proc.Gener. Transm. Distrib.*, vol. 144, no. 1, pp. 6874, 1997.

[31] H. M. Al-Hamadi and S. A. Soliman, "Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model," *Elect. Power Syst. Res.*, vol. 68, no. 1, pp. 4759, Jan. 2004.

[32] Al-Hamadi, H. M., and S. A. Soliman. "Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model." *QNRS Repository* 2011.1 (2011).

[33] Yun, Zhang, et al. "RBF neural network and ANFIS-based short-term load forecasting approach in real-time price environment." *Power Systems, IEEE Transactions on* 23.3 (2008): 853-858.

[34] Xiao, Zhi, et al. "BP neural network with rough set for short term load forecasting." *Expert Systems with Applications* 36.1 (2009): 273-279.

[35] Kandil, Nahi, et al. "An efficient approach for short term load forecasting using artificial neural networks." *International Journal of Electrical Power and Energy Systems* 28.8 (2006): 525-530.

[36] Kandil, Nahi, Vijay Sood, and Maarouf Saad. "Use of ANNs for short-term load forecasting." *Electrical and Computer Engineering, 1999 IEEE Canadian Conference on. Vol. 2. IEEE*, 1999.

[37] Chen, Ying, et al. "Short-term load forecasting: similar day-based wavelet neural networks." *Power Systems, IEEE Transactions on* 25.1 (2010): 322-330.

[38] Hippert, Henrique Steinherz, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. "Neural networks for short-term load forecasting: A review and evaluation." *IEEE Transactions on Power Systems* 16.1 (2001): 44-55.

[39] Osman, Zainab H., Mohamed L. Awad, and Tawfik K. Mahmoud. "Neural network based approach for short-term load forecasting." *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES. IEEE, 2009*.

[40] Vapnik, Vladimir. *The nature of statistical learning theory*. Springer science and business media, 2013.

[41] Schlkopf, Bernhard, and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[42] Bishop, Christopher M. *Pattern recognition and machine learning.* Vol. 4. No. 4. New York: springer, 2006.

[43] Steinwart, Ingo, and Andreas Christmann. *Support vector machines*. Springer Science and Business Media, 2008.

[44] Niu, Dong-xiao, et al. "Study on Forecasting Approach to Short-term Load of SVM Based on Data Mining." *Zhongguo Dianji Gongcheng Xuebao(Proceedings of the Chinese Society of Electrical Engineering)*. Vol. 26. No. 18. 2006.

[45] Hong, Wei-Chiang. "Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm." *Energy* 36.9 (2011): 5568-5578.

[46] Hong, Wei-Chiang. "Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model." *Energy Conversion and Management* 50.1 (2009): 105-117.

[47] Wang, Jianjun, et al. "An annual load forecasting model based on support vector regression with differential evolution algorithm." *Applied Energy* 94 (2012): 65-70.

[48] d'Alche-Buc, Liva Ralaivola Florence. "Dynamical modeling with kernels for nonlinear time series prediction." *Advances in Neural Information Processing Systems 16: Proceedings of the 2003 Conference*. Vol. 16. MIT Press, 2004.

[49] Ralaivola, Liva, and Florence D'Alch-Buc. "Time series filtering, smoothing and learning using the kernel Kalman filter." *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on. Vol. 3. IEEE*, 2005.

[50] Che, JinXing, and JianZhou Wang. "Short-term load forecasting using a kernel-based support vector regression combination model." *Applied Energy* 132 (2014): 602-609.

[51] Sapankevych, Nicholas I., and Ravi Sankar. "Time series prediction using support vector machines: a survey." *Computational Intelligence Magazine*, IEEE 4.2 (2009): 24-38.

[52] Wang, Bo, et al. "A new ARMAX model based on evolutionary algorithm and particle swarm optimization for short-term load forecasting." *Electric Power Systems Research* 78.10 (2008): 1679-1685.

[53] Hinojosa, V. H., and A. Hoese. "Short-term load forecasting using fuzzy inductive reasoning and evolutionary algorithms." *IEEE Transactions on Power Systems* 25.1 (2010): 565-574.

[54] Fan, Shu, and Rob J. Hyndman. "Short-term load forecasting based on a semi-parametric additive model." *IEEE Transactions on Power Systems* 27.1 (2012): 134-141.

[55] Jin, Min, et al. "Short-term power load forecasting using grey correlation contest modeling." *Expert Systems with Applications* 39.1 (2012): 773-779.

[56] Willis, H. Lee. "Analytical methods and rules of thumb for modeling DG-distribution interaction." *Power Engineering Society Summer Meeting, 2000. IEEE*. Vol. 3. IEEE, 2000.

[57] Lee, Soo-Hyoung, and Jung-Wook Park. "Selection of optimal location and size of multiple distributed generations by using Kalman filter algorithm." *IEEE Transactions on Power Systems* 24.3 (2009): 1393-1400.

[58] Gzel, Tuba, and M. Hakan Hocaoglu. "An analytical method for the sizing and siting of distributed generators in radial systems." *Electric Power Systems Research* 79.6 (2009): 912-918.

[59] Keane, Andrew, and Mark O'Malley. "Optimal allocation of embedded generation on distribution networks." *IEEE Transactions on Power Systems* 20.3 (2005): 1640-1646.

[60] Singh, Amit K., and S. K. Parida. "Optimal placement of DGs using MINLP in deregulated electricity market." *Energy and Sustainable Development: Issues and Strategies (ESD), 2010 Proceedings of the International Conference on. IEEE*, 2010.

[61] Khalesi, N., N. Rezaei, and M-R. Haghifam. "DG allocation with application of dynamic programming for loss reduction and reliability improvement." *International Journal of Electrical Power and Energy Systems* 33.2 (2011): 288-295.

[62] AlHajri, Mohamad F., Mohammed R. AlRashidi, and Mohamed E. El-Hawary. "Improved Sequential Quadratic Programming approach for optimal Distribution Generation sizing in distribution networks." *Electrical and Computer Engineering (CCECE), 2010 23rd Canadian Conference on. IEEE*, 2010.

[63] Jabr, R. A., and B. C. Pal. "Ordinal optimisation approach for locating and sizing of distributed generation." *IET generation, transmission and distribution* 3.8 (2009): 713-723.

[64] Singh, R. K., and S. K. Goswami. "Optimum siting and sizing of distributed generations in radial and networked systems." *Electric Power Components and Systems* 37.2 (2009): 127-145.

[65] Singh, Deependra, and K. S. Verma. "Multiobjective optimization for DG planning with load models." *IEEE Transactions on Power Systems* 24.1 (2009): 427-436.

[66] Aly, Akram I., Yasser G. Hegazy, and Metwally A. Alsharkawy. "A simulated annealing algorithm for multi-objective distributed generation planning." *IEEE Power and Energy Society General Meeting*. 2010.

[67] El-Zonkoly, A. M. "Optimal placement of multi-distributed generation units including different load models using particle swarm optimization." *Swarm and Evolutionary Computation* 1.1 (2011): 50-59.

[68] Hejazi, Hosein A., et al. "Distributed generation site and size allocation through a techno economical multi-objective Differential Evolution Algorithm." *Power and Energy (PECon), 2010 IEEE International Conference on. IEEE*, 2010.

[69] Wang, Lingfeng, and Chanan Singh. "Reliability-constrained optimum placement of reclosers and distributed generators in distribution networks using an ant colony system algorithm." *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 38.6 (2008): 757-764.

[70] Abu-Mouti, Fahad S., and M. E. El-Hawary. "Optimal distributed generation allocation and sizing in distribution systems via artificial bee colony algorithm." *IEEE Transactions on Power Delivery* 26.4 (2011): 2090-2101.

[71] Nara, Koichi, et al. "Application of tabu search to optimal placement of distributed generators," *Power Engineering Society Winter Meeting*, 2001. IEEE. Vol. 2. IEEE, 2001.

[72] Niknam, T., et al. "Optimal operation of distribution system with regard to distributed generation: a comparison of evolutionary methods." *Industry Applications Conference, 2005*. Fourtieth IAS Annual Meeting. Conference Record of the 2005. Vol. 4. IEEE, 2005.

[73] Sulaiman, M. H., et al. "Optimal allocation and sizing of distributed generation in distribution system via firefly algorithm." *Power Engineering and Optimization Conference (PEDCO) Melaka, Malaysia, 2012 Ieee International. IEEE*, 2012.

[74] Yang, Xin-She, and Suash Deb. "Engineering optimisation by cuckoo search." *International Journal of Mathematical Modelling and Numerical Optimisation* 1.4 (2010): 330-343.

[75] Moravej, Zahra, and Amir Akhlaghi. "A novel approach based on cuckoo search for DG allocation in distribution network." *International Journal of Electrical Power and Energy Systems* 44.1 (2013): 672-679.

[76] Atashpaz-Gargari, Esmaeil, and Caro Lucas. "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition." *Evolutionary computation*, 2007. CEC 2007. IEEE Congress on. IEEE, 2007.

[77] Soroudi, Alireza, and Mehdi Ehsan. "Imperialist competition algorithm for distributed generation connections." *Generation, Transmission and Distribution, IET* 6.1 (2012): 21-29.

[78] Gandomkar, M., M. Vakilian, and M. Ehsan. "A genetic-based tabu search algorithm for optimal DG allocation in distribution networks." *Electric power components and systems* 33.12 (2005): 1351-1362.

[79] Soroudi, Alireza, and Mehdi Ehsan. "Efficient immune-GA method for DNOs in sizing and placement of distributed generation units." *European Transactions on Electrical Power* 21.3 (2011): 1361-1375.

[80] Ramirez-Rosado, Igancio J., and J. Antonio Domnguez-Navarro. "Possibilistic model based on fuzzy sets for the multiobjective optimal planning of electric power distribution networks." *IEEE Transactions on Power Systems* 19.4 (2004): 1801-1810.

[81] AlHajri, M. F., M. R. AlRashidi, and M. E. El-Hawary. "Hybrid particle swarm optimization approach for optimal distribution generation sizing and allocation in distribution systems." *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on. IEEE*, 2007.

[82] Tan, Wen-Shan, et al. "Optimal distributed renewable generation planning: A review of different approaches." *Renewable and Sustainable Energy Reviews* 18 (2013): 626-645.

[83] Singh, Rayman Preet, Peter Xiang Gao, and Daniel J. Lizotte. "On hourly home peak load prediction." *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on. IEEE*, 2012.

[84] Son, Sung-Yong, et al. "Feature selection for daily peak load forecasting using a neuro-fuzzy system." *Multimedia Tools and Applications* 74.7 (2015): 2321-2336.

[85] "Backcasted Actual Load Profiles - Historical." http://www.ercot.com/mktinfo/loadprofile/alp.

[86] Nagi, J., Yap, K. S., Nagi, F., Tiong, S. K., and Ahmed, S. K. (2011). "A computational intelligence scheme for the prediction of the daily peak load," *Applied Soft Computing*, 11(8), 4773-4788.

[87] Moazzami, M., A. Khodabakhshian, and R. Hooshmand. "A new hybrid day-ahead peak load forecasting method for Irans National Grid." *Applied Energy* 101 (2013): 489-501.

[88] National Weather Service Forecast Office, *NWSFO* [Online]. Available: http://www.nws.noaa.gov/climate/index.php

[89] Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. Springer, Berlin: Springer series in statistics, 2001.

[90] Rech, Gianluigi, Timo Tersvirta, and Rolf Tschernig. "A simple variable selection technique for nonlinear models." *Communications in Statistics-Theory and Methods* 30.6 (2001): 1227-1241.

[91] Chipman, Hugh A., Edward I. George, and Robert E. McCulloch. "BART: Bayesian additive regression trees." *The Annals of Applied Statistics* (2010): 266-298.

[92] Chipman, Hugh, Pritam Ranjan, and Weiwei Wang. "Sequential design for computer experiments with a flexible Bayesian additive model." *Canadian Journal of Statistics* 40.4 (2012): 663-678.

[93] Zou, Hui, and Trevor Hastie. "Regularization and variable selection via the elastic net." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2 (2005): 301-320.

[94] Friedman, Jerome, Trevor Hastie, and Rob Tibshirani. "Regularization paths for generalized linear models via coordinate descent." *Journal of statistical software* 33.1 (2010): 1.

[95] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[96] Peng, Hanchuan, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005): 1226-1238.

[97] Rasmussen, Carl Edward. "Gaussian processes for machine learning." (2006).

[98] Bleich, Justin, et al. "Variable selection for BART: An application to gene regulation." *The Annals of Applied Statistics* 8.3 (2014): 1750-1781.

[99] Kapelner, Adam, and Justin Bleich. "bartMachine: Machine Learning with Bayesian Additive Regression Trees." arXiv preprint arXiv:1312.2171 (2013).

[100] Bakirtzis, A. G., et al. "Short term load forecasting using fuzzy neural networks." *IEEE Transactions on Power Systems* 10.3 (1995): 1518-1524.

[101] Fernndez-Blanco, Ricardo, Jose M. Arroyo, and Natalia Alguacil. "Network-constrained day-ahead auction for consumer payment minimization," *IEEE transactions on Power Systems* 29.2 (2014): 526-536.

[102] Clements, A. E., A. S. Hurn, and Zili Li. "Forecasting day-ahead electricity load using a multiple equation time series approach." *European Journal of Operational Research* 251.2 (2016): 522-530.

[103] Moghram, Ibrahim, and Saifur Rahman. "Analysis and evaluation of five short-term load forecasting techniques." *IEEE Transactions on Power Systems* 4.4 (1989): 1484-1491.

[104] Huang, Shyh-Jier, and Kuang-Rong Shih. "Short-term load forecasting via ARMA model identification including non-Gaussian process considerations." *IEEE Transactions on Power Systems* 18.2 (2003): 673-679.

[105] Papadopoulos, Sokratis, and Ioannis Karakatsanis. "Short-term electricity load forecasting using time series and ensemble learning methods." *Power and Energy Conference at Illinois (PECI)*, 2015 IEEE. IEEE, 2015.

[106] Hippert, Henrique Steinherz, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. "Neural networks for short-term load forecasting: A review and evaluation." *IEEE Transactions on Power Systems* 16.1 (2001): 44-55.

[107] Hernandez, Luis, et al. "A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings." *IEEE Communications Surveys and Tutorials* 16.3 (2014): 1460-1495.

[108] Pandey, Ajay Shekhar, Devender Singh, and Sunil Kumar Sinha. "Intelligent hybrid wavelet models for short-term load forecasting." *IEEE Transactions on Power Systems* 25.3 (2010): 1266-1273.

[109] Paparoditis, Efstathios, and Theofanis Sapatinas. "Short-term load forecasting: the similar shape functional time-series predictor." *IEEE Transactions on Power Systems* 28.4 (2013): 3818-3825.

[110] Chen, Bo-Juen, and Ming-Wei Chang. "Load forecasting using support vector machines: A study on EUNITE competition 2001." *IEEE Transactions on Power Systems* 19.4 (2004): 1821-1830.

[111] Sapankevych, Nicholas I., and Ravi Sankar. "Time series prediction using support vector machines: a survey." *IEEE Computational Intelligence Magazine* 4.2 (2009): 24-38.

[112] MacKay, David JC. "Introduction to Gaussian processes." *NATO ASI Series F Computer and Systems Sciences* 168 (1998): 133-166.

[113] Mori, Hiroyuki, and Masatarou Ohmi. "Probabilistic short-term load forecasting with Gaussian processes." *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on. IEEE*, 2005.

[114] Mori, Hiroyuki, and Daisuke Kanaoka. "GP-based temperature forecasting for electric load forecasting." *TENCON 2009-2009 IEEE Region 10 Conference*. IEEE, 2009.

[115] Yan, Junchi, et al. "Load forecasting using twin gaussian process model." Service Operations and Logistics, and Informatics (SOLI), *2012 IEEE International Conference on. IEEE*, 2012.

[116] Alamaniotis, Miltiadis, Stylianos Chatzidakis, and Lefteri H. Tsoukalas. "Monthly load forecasting using kernel based gaussian process regression." *MedPower 2014. IET*, 2014.

[117] Lee, Duehee, and Ross Baldick. "Short-term wind power ensemble prediction based on Gaussian processes and neural networks." *IEEE Transactions on Smart Grid* 5.1 (2014): 501-510.

[118] Alamaniotis, Miltiadis, Andreas Ikonomopoulos, and Lefteri H. Tsoukalas. "Evolutionary multiobjective optimization of kernel-based very-short-term load forecasting." *IEEE Transactions on Power Systems* 27.3 (2012): 1477-1484.

[119] Skretting, Karl, and Kjersti Engan. "Recursive least squares dictionary learning algorithm." *IEEE Transactions on Signal Processing* 58.4 (2010): 2121-2130.

[120] Aharon, Michal, Michael Elad, and Alfred Bruckstein. "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation." *IEEE Transactions on Signal Processing* 54.11 (2006): 4311-4322.

[121] Chen, Tairen, et al. "Distribution-level peak load prediction based on Bayesian Additive Regression Trees." Power and Energy Society General Meeting (PESGM), 2016. IEEE, 2016.

[122] A. Y. Moustafa and E. F. El-Saadany. "Optimal allocation of ESS in distribution systems with a high penetration of wind energy," *IEEE Transactions on Power Systems*, vol. 25, no. 4, pp. 1815-1822, 2010.

[123] T. Gozel and M. H. Hocaoglu, "An analytical method for the sizing and siting of distributed generators in radial systems," *Electric Power Systems Research*, vol. 79, no. 6, pp. 912-918, 2009.

[124] S. H. Lee and J. W. Park, "Selection of optimal location and size of multiple distributed generations by using Kalman filter algorithm," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1393-1400, 2009.

[125] E. Walid, Y. G. Hegazy, and M. M. A. Salama, "An integrated distributed generation optimization model for distribution system planning," *IEEE Transactions on Power Systems*, vol. 20, no. 2, pp. 1158-1165, 2005.

[126] P. S. Georgilakis and N. D. Hatziargyriou, "Optimal distributed generation placement in power distribution networks: Models, methods, and future research," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3420-3428, 2013.

[127] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.

[128] Wang, H., Murillo-Sanchez, C. E., Zimmerman, R. D., and Thomas, R. J. (2007). "On computational issues of market-based optimal power flow." *IEEE Transactions on Power Systems*, 22(3), 1185-1193.

[129] Hwang, C-L., and Abu Syed Md Masud. "Multiple objective decision makingmethods and applications: a state-of-the-art survey." Vol. 164. *Springer Science and Business Media*, 2012.

[130] Marler, R. Timothy, and Jasbir S. Arora. "Survey of multi-objective optimization methods for engineering." *Structural and multidisciplinary optimization* 26.6 (2004): 369-395.

[131] Konak, A., Coit, D. W., and Smith, A. E. (2006). "Multi-objective optimization using genetic algorithms: A tutorial." *Reliability Engineering and System Safety*, 91(9), 992-1007.

[132] H. D. Chiang and J. J. Rene, "Toward a practical performance index for predicting voltage collapse in electric power systems," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 584-592, 1995.

[133] R. S. Al Abri, E. F. El-Saadany, and Y. M. Atwa, "Optimal placement and sizing method to improve the voltage stability margin in a distribution system using distributed generation," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 326-334, 2013.

[134] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12-19, 2011.

[135] Christie, Richard D. "Power systems test case archive." *Electrical Engineering Department, University of Washington* (2000).

[136] Sivanandam, S. N., Sai Sumathi and S. N. Deepa. *Introduction to fuzzy logic using MATLAB. Vol. 1*. Berlin: Springer, 2007.

[137] K. Deb and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling," in *Evolutionary Multi-Criterion Optimization*, Berlin, Heidelberg: Springer, 2007, pp. 803-817.

[138] Wilson, A. G., and Adams, R. P. (2013, February). ”Gaussian Process Kernels for Pattern Discovery and Extrapolation.” In *ICML* (3) (pp. 1067-1075).